الجمهورية الجزائرية الديمقراطية الشعبية

16004.17

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique

Mémoire de Fin d'études Master

Filière : Informatique

Option : Systèmes Informatiques

Thème :

Méthodes de clustering des séquences biologiques pour l'optimisation de l'alignement

Encadré Par :

Lebsir Rabeh

Présenté par :

Bezzazi Haroun Errachid

Juillet 2019

Remerciement

J'adresse mon premier remerciement au bon Dieu, qui nous a donné la volonté, le courage et la patience pour mener à bien ce travail.

Je voudrais tout d'abord adresser toute ma gratitude au directeur de ce mémoire, monsieur rabeh lebsir, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion.

Je remercie mes très chers parents, qui ont toujours été là pour moi. Je remercie mes sœurs pour leurs encouragements.

Enfin, je remercie mes amis qui ont toujours été là pour moi. Leur soutien inconditionnel et leurs encouragements ont été d'une grande aide.

Résumé

Dans ce mémoire de fin d'étude, nous concentrons sur le clustering comme étape préalable à l'alignement des séquences biologiques.

L'objectif de cette étude est d'avoir un bon compromis entre le temps d'exécution et la qualité de l'alignement pour le cas de traitement de masses très importantes de données biologiques.

Nous proposons une stratégie inspirée du paradigme diviser pour conquérir pour améliorer le temps d'exécution en utilisant le k-means comme algorithme de clustering pour le regroupement des séquences biologiques, puis un autre algorithme pour construire l'alignement multiple des séquences (MSA).

L'algorithme k-means nécessite deux paramètres, le nombre voulu de clusters ainsi que les distances entre les individus, dans notre travail, on a introduit une autre notion de distance en utilisant trois paramètres à savoir, l'alignement local, l'alignement global et l'alignement multiple.

Nous avons testé notre approche sur un processeur multi-cœurs avec un ensemble de benchmarks connus dans la littérature. Les résultats montrent que notre approche donne les meilleurs résultats en termes de temps de calcul par rapport aux techniques les plus utilisées, tout en perdant une légère précision.

Mot clés : Alignement multiple de séquences, Clustering, K-means.

Contenu

Liste des figures

Chapitre 1 : L'alignement multiple des séquences

Chapitre 2 : clustring de séquence

Chapitre 3 : notre approche

Chapitre 4 : implémentation

Liste des tableaux

Chapitre 1 : L'alignement multiple des séquences

Chapitre 3 : Notre approche

Introduction général

L'alignement multiple des séquences (MSA) est une tâche très importante dans la bioinformatique. Il permet de représenter deux ou plusieurs séquences de macromolécules biologiques (ADN, ARN ou protéines) l'une sous l'autre, pour faire ressortir les régions similaires ou homologues.

MSA est utilisé pour des tâches complexes telles que l'analyse des protéines, l'identification des sites fonctionnels dans les séquences génomiques, la prédiction structurelle et fonctionnelle. Malheureusement, faire un alignement multiple précis a été montré NP-Compliqué[1]. Par conséquent, MSA est un problème d'optimisation qui présente une grande complexité en temps et en espace. En conséquence, plusieurs méthodes ont été proposées pour faire face à ce problème. Ils peuvent être regroupés en trois classes[2]

La première classe comprend les méthodes exactes qui utilisent une généralisation de l'algorithme Needleman [3] pour aligner toutes les séquences simultanément. Bien que les méthodes exactes fournissent des solutions optimales, mais leur principal inconvénient est leur complexité en temps et en espace, ce qui devient encore plus critique avec le nombre et la longueur accrus des séquences et devient rapidement inutilisable.

La deuxième classe contient des méthodes basées sur une approche progressive[4]. L'alignement progressif crée un MSA final en combinant les alignements par paires, en commençant par la paire la plus semblable et en progressant vers les relations les plus éloignées. Les méthodes progressives sont simples, rapides et généralement donnent de bonnes qualités d'alignements. Cependant, leur principal inconvénient est le problème des minimums locaux et, par conséquent, ils peuvent conduire à de mauvaises solutions.

La troisième classe se compose de méthodes itératives. L'idée de base est de commencer par un alignement initial et d'affiner de façon itérative grâce à une série d'améliorations appropriées appelées itérations. Le processus est répété jusqu'à la satisfaction de certains critères. Ces méthodes ont été prometteuses mais peuvent être beaucoup plus lentes, ce qui peut les rendre inutilisables pour les données à grande échelle.

Avec l'apparition des bases de données volumineuses et qui contiennent des masse très importantes de données, la plus part de ces méthodes sont devenues obsolètes.De nouvelles idées sont apparues, on trouve par exemple l'application du clustering des séquences afin de minimiser la complexité du problème, donc, au lieu d'aligner un ensemble volumineux de séquences qui nécessité dans la plus part des cas une comparaison deux à deux, l'idée est de décomposer l'ensemble en sous-ensembles afin de les traiter séparément, ce qui va améliorer considérablement le temps d'exécution. La difficulté principale de ces méthodes est le choix de l'algorithme de clustering.

Dans notre étude, on a proposé d'intégrer l'algorithme k-means dans une phase préalable afin de générer des sous-ensembles de qualité et d'avoir à la fin un bon compromis Temps/qualité.

1. L'alignement multiple de séquences

1.1. Introduction

L'alignement multiple des séquences consiste en l'écriture de plusieurs séquences d'une façon superposée afin de faire des analyses approfondies, comme l'analyse des familles de protéines, la compréhension de leurs tendances évolutionnaires et la détection des homologies. A partir d'un bon alignement, on peut facilement extraire des informations sur les origines de ces séquences, la modélisation, la structure, etc.

Donc, l'alignement multiple de séquences MSA (Multiple Sequence Alignment) consiste à aligner plusieurs séquences d'une façon globale afin de tirer les relations entre une famille de séquences (Figure 1.1). Le but principal de l'alignement multiple des séquences biologiques est de montrer les caractéristiques communes entre un ensemble de séquences de protéines ou de nucléotides. Le MSA permet de caractériser les régions conservées et les régions variables au sein d'une famille de séquences (Figure1.2). Il permet aussi de construire la séquence consensus de

plusieurs séquences alignées. Pour une meilleure compréhension de l'évolution des séquences biologiques, le MSA pourra contribuer efficacement. L'alignement multiple est également utilisé dans plusieurs autres domaines comme la bioinformatique structurelle où le MSA est utilisé pour la prédiction structurelle et fonctionnelle des protéines.

Dans ce chapitre, on a choisi de suivre la référence [5] dans sa présentation de l'état de l'art vu qu'elle est bien présenté par l'auteur.

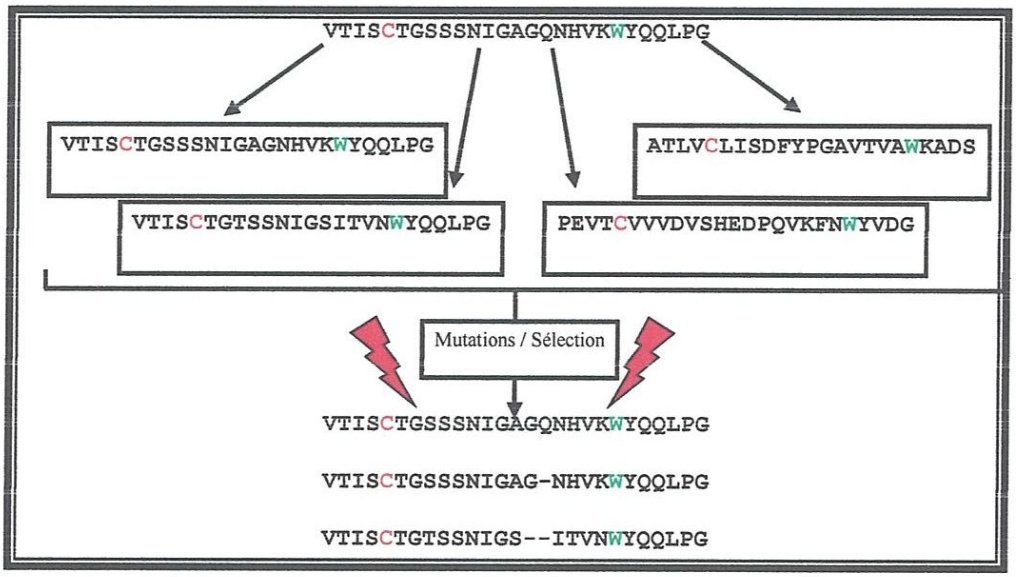

***Figure 1. 1** Alignement multiple : une histoire[5]*

Vu la quantité importante des données biologiques produisces par des projets les différents projets tels que le projet Genome Humain, 1000 Genomes, HapMap, etc. la construction des alignements d'une façon manuelle est devenue une tâche non praticable, par conséquent, des méthodes automatiques sont proposés. La construction automatique des alignements est devenue aujourd'hui une tâche importante en bioinformatique.

En effet, pour aligner 3 séquences de taille 1000, il faut 10003 soit 1 Go de mémoire alors que les biologistes ont souvent plusieurs centaines de séquences à aligner et ils veulent des solutions optimales dans des courtes durées. Donc, le problème d'alignement multiple est plus complexe qu'une simple et directe généralisation d'alignement de paires de séquences.[5]

Afin de résoudre un problème d'alignement des séquences, il faut connaitre le type des séquences à aligner, une chose qui rentre dans la sémantique des séquences biologiques, puis on doit avoir un algorithme par lequel on va faire l'alignement et à la fin, on doit avoir une technique par laquelle on peut prendre une décision sur la qualité de l'alignement.

Afin de faire un bon alignement, le biologiste doit faire un bon choix de séquences, l'intégration d'une séquence orpheline dans un alignement multiple de séquences peut donner des résultats indésirables. Aussi, le choix d'un algorithme doit être prudent, la taille des séquences peut rendre l'utilisation de certains algorithmes obsolète.

La validation des algorithmes d'alignement passe par deux étape ; la première est le choix d'une fonction objective, cette fonction permet de choisir le meilleur alignement parmi d'autre, cette étape se déroule pendant l'alignement. Une autre mesure, c'est la comparaison de résultat de

ces alignements avec des alignements manuelles, pour cela, des Benchmarks ont étaient proposés tel que BaliBase. Ces dernières contiennent des familles de séquences dont l'alignement multiple optimal (du point de vue biologique) est connu et généralement crée à la main. Le troisième problème lié à l'alignement multiple est calculatoire

Les solutions proposées pour l'alignement multiples des séquences biologiques peuvent être classés en trois catégories principales à savoir, exactes, progressives et itératives.

Les algorithmes exacts sont une généralisation de l'algorithme Needlman&Wansch, et donc fournissent habituellement un alignement très près de l'optimalité. Néanmoins, elles sont limitées et deviennent rapidement obsolètes dans le cas de l'alignement de séquences de grandes tailles ou dans le cas du nombre des séquences dépassant les dix séquences.

Les méthodes progressives à l'encontre des méthodes exactes ne garantissent pas une meilleure solution. Elles construisent un alignement progressivement en alignant les séquences deux à deux jusqu'à l'obtention de l'alignement global. Ces méthodes sont rapides et simple ce qui a rendu leur utilisation très répondue, par contre, une étape préalable doit définir l'ordre de l'alignement, c'est bien la construction d'un arbre phylogénétique.

Troisièmement, les méthodes d'alignement itératives utilisent un alignement provisoire peut être généré par n'importe quelle méthode rapide et ensuite la raffine par une série de raffinements itératifs jusqu'il n'y aura plus d'améliorations qui peuvent être apportées.

Dans ce qui suit, nous présentons en détail les arbres phylogénétiques qui sont très utilisés dans les méthodes d'alignement progressives. Ils permettent de faire le choix et le tri des séquences à aligner.

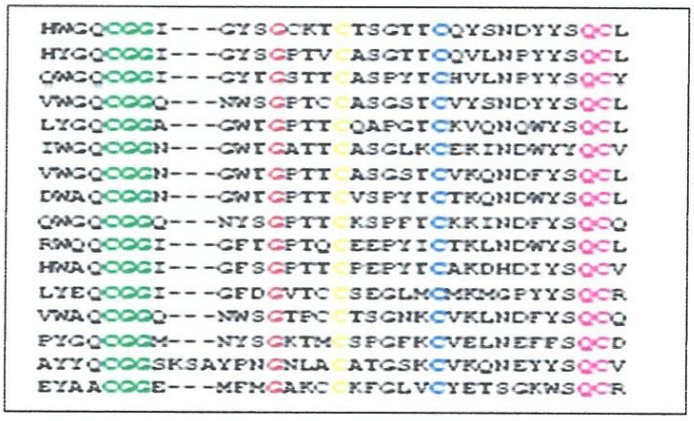

Figure 1. 2 *Site de fixation de la cellulose (extrait de prosite entrée PS00562).[5]*

1.2. Les approches de résolution du problème MSA

Vu que la recherche d'un alignement optimal de séquences multiples se caractérise par une complexité temporelle et spatiale accrue, de nombreuses méthodes heuristiques ont été développées pour trouver des solutions quasi optimales dans des délais d'exécution courts.

Ces méthodes peuvent être décomposées en trois grandes familles à savoir, les méthodes exactes, les méthodes progressives et les méthodes itératives.

1.2.1. Les méthodes exactes

Les algorithmes exacts ont été développés pour aligner plusieurs séquences simultanément. Ils sont des heuristiques de haute qualité capables de produire des alignements proches de ceux optimaux, mais elles sont limitées au traitement d'un petit nombre de séquences. Ainsi, des besoins importants en mémoire, un temps de calcul important et une limitation du nombre de séquences limitent leur utilisation.

En fait, il s'agit d'une généralisation de l'algorithme de programmation dynamique de Needleman-Wunsh développé pour l'alignement de deux séquences, pour l'alignement multiple de n séquences en utilisant une matrice de scores à n dimensions. Par conséquent, pour L séquences de longueur N, la taille de la matrice est L^N. Cette approche est tellement gourmande en ressources qu'elle devient impraticable pour le cas des données de grandes tailles.

La *(Figure1.3)* montre un exemple de table de score d'un alignement de trois séquences ainsi que le chemin optimal qui donne l'alignement idéal. [5]

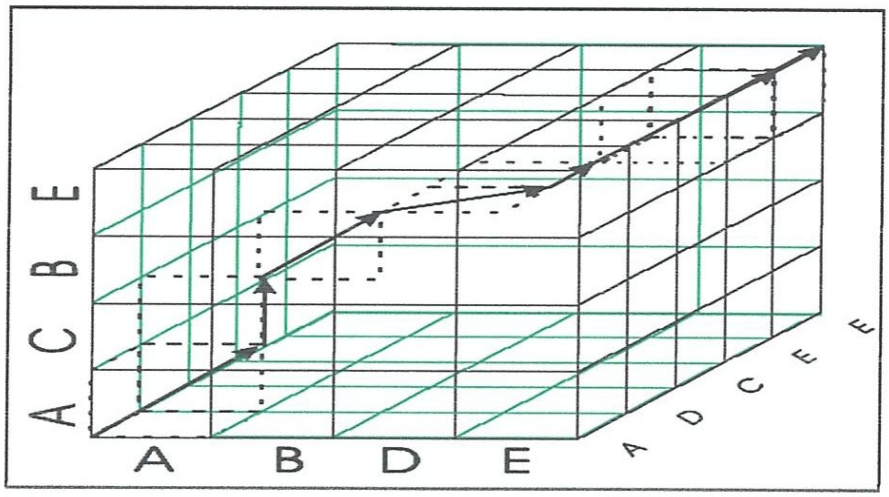

Figure 1. 3 : *La trace back dans un alignement de trois séquences. [5]*

Les algorithmes basés sur les approches exactes et ainsi sur la programmation dynamique, à savoir l'alignement local de deux séquences en utilisant l'algorithme de Smith&Waterman ou l'alignement global en utilisant l'algorithme de Needlman&Wansch representent une complexité temporelle et spatiale importante ce qui les rend rapidement obsolète.

1.2.2. Les méthodes progressives

Vu la complexité des méthodes exactes, plusieurs heuristiques ont été proposées pour surmonter l'impraticabilité de ces méthodes. Parmi les méthodes les plus répandues et les plus populaires en MSA, on trouve les méthodes progressives. Ces méthodes sont simples, rapides et donnent généralement des solutions acceptables. L'algorithme de l'alignement progressive a été premièrement décrit par Hogeweg [6] et plus tard redéfini par Feng[7]. Les méthodes d'alignement multiples les plus employés couramment sont basés sur l'exécution de cet algorithme comme la méthode CLUSTAL W [8] qui est considérée comme la méthode standard d'alignement multiple.

Les méthodes progressives sont les méthodes les plus populaires [7]. Ces méthodes d'alignement progressif appliquent de manière répétitive des algorithmes d'alignement par paires pour créer un alignement multiple de séquences. Ces méthodes sont simples, rapides et ne nécessitent pas une mémoire importante. Pour n séquence de longueur L, le temps d'exécution pour l'alignement progressif est O (nL2) [1]. Fondamentalement, les principales étapes d'un algorithme progressif sont les suivantes:

1. Choisir deux séquences et aligner-les.
2. Choisir une autre séquence et aligner-la avec les séquences précédemment alignées.

3. Répéter l'étape 2 jusqu'à ce que toutes les séquences soient alignées.

Cependant, le problème dans ces méthodes est de déterminer l'ordre selon lequel les séquences sont alignées. L'une des solutions les plus utilisées est l'arbre de phylogénétique proposé par Feng et Doolittle [4]. Cet arbre donne l'ordre dans lequel l'alignement progressif sera effectué. Après l'alignement des deux séquences les plus proches, plusieurs séquences sont ajoutées en les alignant avec l'alignement existant. Le suivi d'un arbre phylogénétique ne donne pas nécessairement un alignement optimal, même si l'arbre est parfait. Par exemple, les erreurs commises aux premières étapes sont propagées aux dernières étapes et ne peuvent pas être corrigées. Une variété de programmes MSA sont basés sur les algorithmes Feng et Doolittle tels que CLUSTAL, MULTALIGN, etc. [7]. La seule différence entre ces programmes réside dans la stratégie utilisée pour créer l'arbre guide ou l'arbre phylogénétique; Par exemple, CLUSTAL utilise l'algorithme de jonction de voisin [8]. En règle générale, l'algorithme de base de Feng-Doolittle est composé des étapes suivantes:

1. Calculer tous les alignements par paires possibles entre les séquences (toutes les combinaisons des séquences).

2. Convertir les scores des alignements en distances et calculez l'arbre guide en utilisant la matrice de distance et une méthode de construction d'arbre.

3. Un alignement multiple est créé progressivement en commençant par les séquences les plus proches. Les séquences sont ajoutées une par une en fonction de l'ordre donné par l'arbre guide.

Les méthodes progressives peuvent être globales ou locales en fonction de l'algorithme d'alignement par paires utilisé. Dans le cas d'un alignement progressif global tel que CLUSTAL [8], un algorithme global par paire tel que l'algorithme de Needleman [3] est utilisé pour aligner chaque séquence non alignée sur les séquences alignées précédemment dans les étapes précédentes. Cependant, l'alignement multiple local utilise un algorithme de programmation dynamique local tel que l'algorithme de Smith-Waterman pour aligner uniquement les motifs les plus conservés. Il a été démontré que les algorithmes globaux ont beaucoup de succès dans les cas impliquant des séquences équidistantes, des familles divergentes de séquences et l'alignement de séquences orphelines avec une famille. Cependant, ils sont moins précis en présence de grandes extensions N-C-terminales et d'insertions internes que les méthodes locales [9].

L'exemple suivant, extrait du [5] montre les étapes de construction d'un alignement multiple en utilisant une simple méthode progressive : Ayant les 5 séquences suivantes :

S1= ATTCGGATT

S2= ATCCGGATT

S3= ATGGAATTTT

S4= ATGTTGTT

S5= AGTCAGG

La méthode d'alignement progressive commence d'abord par l'alignement de deux séquences par exemple S1 et S2 :

```
S1: A T T C G G A T T
S2: A T C C G G A T T
```

Ensuite on ajoute une troisième séquence à l'alignement précèdent (soit S3). S3 sera alignée avec la séquence la plus proche par exemple S1. L'alignement de S1 et S3 insère deux gaps en S1, alors on doit propager les deux gaps dans la séquence S2. Les positions des gaps ajoutés sont conservées définitivement.

```
S2: A T C C G G A T T - -
S1: A T T C G G A T T - -
S3: A T G - G A A T T T T
```

De la même façon on ajoute les séquences S4 et S5.

```
S2: A T C C G G A T T - -
S3: A T G - G A A T T T T
S1: A T T C G G A T T - -
S4: A T G T T G - T T - -
```

Apres l'alignement de S5, on aura le résultat d'alignement multiple de séquences suivant :

```
S2: A T C C G G A T T - -
S3: A T G - G A A T T T T
S4: A T G T T G - T T - -
S1: A T T C G G A T T - -
S5: A G T C A G G - - - -
```

La complexité des méthodes progressives est généralement inférieure à $o(N^3L^2)$ dans le pire des cas, (L est le nombre des séquences et N est la taille de la plus grande séquence). Elle est nettement inférieure à celle des méthodes exactes $o(N^2 2^L N^L)$. Actuellement, La technique d'alignement multiple la plus rapide est la méthode MAFFT (Tableau 1.1). MAFFT utilise un nouvel algorithme pour l'alignement de paires de séquences basé sur la transformation de Fourier. [5]

ALIGNER	**Complexité**
ClustalW	$o(N^2L^2)$
MAFFT	$o(N^2)$
T-COFFEE	$o(N^2L^2)+o(N^3L)+o(N^3)+o(NL^2)$
MUSCLE	$o(N^3L)$

Tableau 1. 1: *La complexité de quelques programmes progressifs*[5]

Clustal W [8] est basé sur le principe de l'approche Feng et Doolittle. Le principe général est constitué de trois grande étape : calculer les distances entre toutes les combinaisons des séquences, à partir de la table des distance, tracer l'arbre phylogénétique et en fin construire le MSA en suivant l'arbre phylogénétique.

Etape1 : calcul des distances

- La similarité de chaque séquence est évaluée par rapport à toutes les séquences.
- Un score de similitude est calculé pour chaque paire de séquences selon un alignement approximatif global rapide. On obtient ainsi une matrice de distances.

Etape2 : construction de l'arbre

- Un arbre phylogénétique est construit : il s'agit d'un arrangement traduisant les relations globales de parenté entre les séquences. Cet arbre phylogénique est construit selon la méthode "Neighbour-Joining".

- Il indique l'ordre à partir duquel l'alignement multiple graduel sera établi.

Etape3 :

- Utilisez l'arbre guide pour déterminer l'ordre dans lequel le séquences doit être alignées :

Les étapes de l'algorithme sont comme suit

a- Choisissez les séquences ou les profils à aligner en suivant l'arbre guide.

b- Les aligner à l'aide d'une méthode basée avec une programmation dynamique.

c- Créer un profil de l'alignement établi à partir du site résultat

d- Si la racine de l'arbre n'est pas atteinte, aller à l'étape a.

Depuis près de vingt ans, Clustal W a été l'algorithme d'alignement de multiples, et il reste encore aujourd'hui très utilisé.

Clustal W peut aligner plus d'une centaine de séquences, et il donne de bons résultats dans la plupart des cas. Néanmoins, son majeur inconvénient réside dans l'étape de construction de l'arbre, qui va devenir rapidement obsolète pour les grandes masses de données.

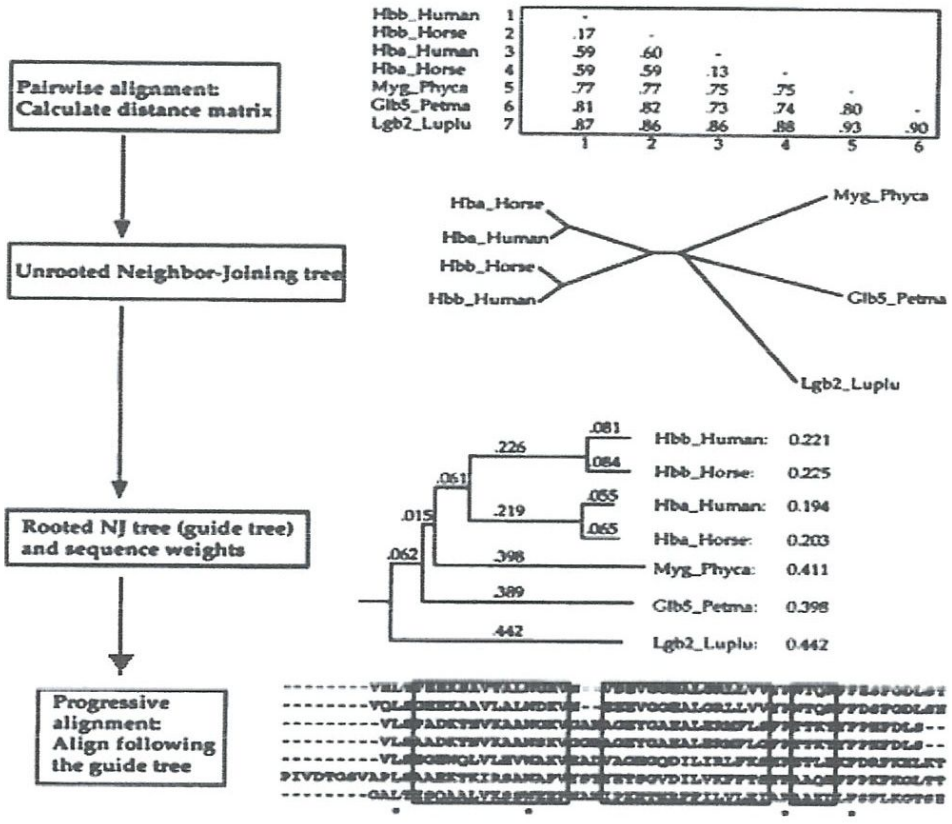

Figure 1. 4 : *le déroulement de programme ClustalW. [5]*

1.2.3. Les méthodes itératives

Les méthodes d'alignement itératives reposent sur des algorithmes capables de produire un alignement initial et de l'affiner au moyen d'une série d'itérations jusqu'à ce qu'aucune amélioration ne puisse être apportée. Ils reposent sur l'idée que la solution peut être élaborée à partir d'une solution sous-optimale déjà existante.

Par exemple, PRRP [10] et MUSCLE [11] optimisent un alignement global progressif en divisant les séquences en deux groupes à l'aide d'un algorithme d'alignement profil à profil. Dialign [12] utilise les informations locales pour guider l'alignement global. HMMT un modèle utilisant un recuit simulé et une programmation dynamique pour créer un alignement sous-optimal.

La plupart de ces programmes offrent la possibilité de les utiliser intégralement ou partiellement pour les intégrer à un autre processus d'alignement.

1.3. Mesurer un alignement multiple

L'alignement multiple des séquences est une notion purement biologique. Trouver une fonction objective à optimiser qui a les mêmes objectifs biologiques est une tâche quasi-impossible. Afin de mesurer un alignement, plusieurs fonctions et mesures ont été proposé.

Nous présentons dans ce qui suit, les fonctions objectifs proposés pour mesurer un alignement multiple de séquences. [5]

1.3.1. La somme des paires

Cette fonction consiste à sommer tous les scores de toutes les combinaisons possibles.

Soit Ai un alignement de K séquences $\{S1, ..., Sk\}$; [5]

$$SP(\mathrm{Ai}) = \sum_{i=1}^{n-1} \sum_{j=i}^{n} sc(S_i, S_j) \qquad (1.1)$$

Avec Sc(*Si Sj*) est le score de l'alignement de la paire des séquences Si et Sj. Ce score peut être calculé par une mesure de distance ou de similitude. On peut utiliser même les scores des alignements deux à deux, comme le score de l'alignement global avec Needlman & Wansch ou l'algorithme de l'alignement local de Smith & waterman.

Exemple : soit l'alignement A* suivant :

S1 : a c - c d b -

S2 : - c - a d b d

S3 : a - b c d a d

Si on considère que la fonction des distances est :

d (x, x) =0, d(x, y)= 1 pour x ≠ y (y compris les gaps)

Le ***SP*** de l'alignement A* = Sc(S1S2) + Sc(S1S3) + Sc(S2S3)

= 3 + 4 + 5 = 12.

1.3.2. Weighted Sum of Pairs (WSP)

Cette fonction consiste à sommer tous les scores intermédiaires résultants des alignements deux à deux, suivant un arbre phylogénétique.

La formule générale de cette fonction est : [5]

$$\sum_{i=1}^{m-1} \sum_{j=i+1}^{m} Wij.score(Si, Sj) \qquad (1.2)$$

1.4. Evaluation de la qualité d'un algorithme MSA

Les méthodes statistiques et les méthodes d'évaluation par les tests de référence (bases des alignements de référence) sont essentiellement deux méthodes pour l'évaluation des alignements. Les méthodes utilisant les bases de référence sont les plus utiliser àfin de mesurer la qualité d'un algorithme d'alignement multiple de séquences.

1.4.1. Les bases de Tests (Benchmarks)

L'utilisation des benchmarks contenant un nombre important de séquences alignées préalablement à la main par des biologistes est la technique d'évaluation la plus utiliser pour mesurer la qualité des alignements produisent par un algorithme. Une comparaison entre les alignements produises par un algorithme et ceux construis à la main doit être effectuée afin de mesurer la qualité de ces algorithmes.

Plusieurs bases d'alignements de référence ont été élaborées comme BAliBASE[13] PREFAB [14], etc.

1.4.2. La base de référence BALiBASE

BAliBASE est spécialement conçu pour servir de ressource d'évaluation permettant de mesurer la qualité des alignements produis avec un algorithme en les comparant avec les séquences de référence. La base de données contient des alignements multiples de séquences de haute qualité, raffinés manuellement, ainsi que des annotations détaillées.

La base fournit des alignements de référence de haute qualité, basés sur des superpositions structurelles 3D. La version 3.0 de BAliBASE inclut de nouveaux scénarios de test plus complexes, qui représentent les véritables problèmes rencontrés lors de l'alignement de grands ensembles de séquences complexes. À l'aide d'un nouveau protocole de mise à jour semi-automatique, le nombre de familles de protéines dans le test de référence a été augmenté et des

tests représentatifs sont désormais disponibles, couvrant la majeure partie de l'espace de repliement de la protéine. Le nombre total de protéines dans BAliBASE est de 6255 séquences. De plus, des séquences complètes sont maintenant fournies pour tous les tests, ce qui représente des cas difficiles pour les programmes d'alignement globaux et locaux.

Référence	Courte (<100 résidus)	Moyenne (200-300 résidus)	Longue (>400 résidus)
Référence 1: séquences équidistantes de longueurs similaires			
V1 (<25% identité)	7	8	8
V2 (20-40% identité)	10	9	10
V3 (>35% identité)	10	10	8
Référence 2: famille versus orpheline	9	8	7
Référence 3: familles équidistantes divergentes	5	3	5
Référence 4: extension N/C-terminal	12		
Référence 5: insertions	12		

***Tableau 1. 2** : Le contenu de la base BALiBASE. [5]*

1.4.3. L'évaluation des alignements par BALIBASE

Deux mesures sont proposées pour connaitre la qualité d'un alignement CS (Column Score) et SPS (Sum of Pairs Score). Le premier pourra être considéré comme étant un score mathématique alors que le deuxième est considéré comme un score biologique.

Le CS consiste en un calcul de moyenne des colonnes correctement alignées. Il peut être calculé comme suit :

$$CS = \frac{\sum_{i=1}^{l} C_i}{l} \qquad (1.10)\ [5]$$

Avec Ci est le score de la colonne i.

D'autre part, le SPS détermine le nombre de résidus correctement alignés par rapport à un alignement considéré comme référence, raffiné à la main par les biologistes.

1.5. Conclusion

Nous avons vu dans ce chapitre les notions de base de l'alignement multiple de séquences. On a présenté la méthode pour résoudre et pour mesurer la qualité d'un MSA et à la fin, comment mesurer la qualité d'un algorithme d'alignement en utilisant des benchmarks

Comme déjà mentionné, nous allons utiliser une nouvelle approche pour traiter le problème de MSA.

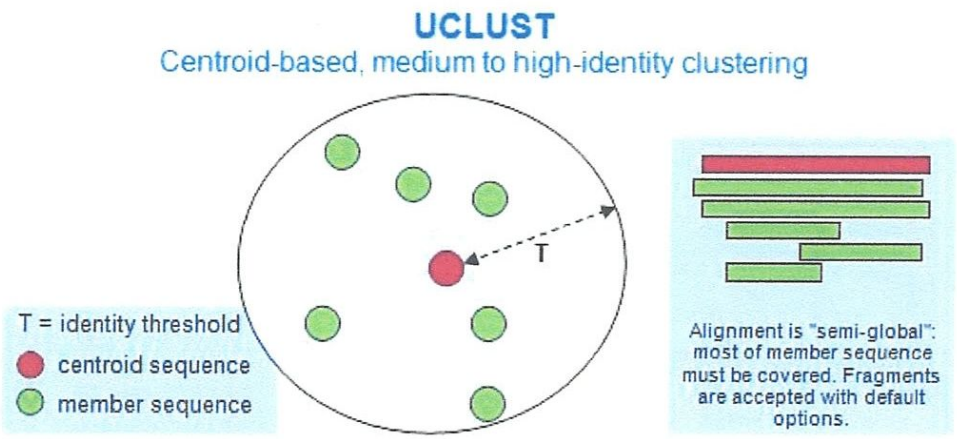

Figure 2. 1 **: l'algorithme UCLUST [17]**

2.3. CD-HIT

CD-HIT est un algorithme de clustering évolutif. L'algorithme de base CD-HIT trie les séquences d'entrée du plus long au plus court et les traite séquentiellement du plus long au plus court. La première séquence est automatiquement classée en tant que première séquence représentative de cluster. Ensuite, chaque séquence d'interrogation des séquences restantes est comparée aux séquences représentatives trouvées avant et est classée comme redondante ou représentative en fonction de son caractère similaire à l'une des séquences représentatives existantes. En mode précis, une requête est comparée à tous les représentants et regroupée au plus proche. [18]D'autres méthodes de clustering sont proposées mais elles souffrent de deux problèmes principaux dans le cas de leurs utilisations dans l'alignement multiple :

Le temps important vu que ces algorithmes suivent une méthode évolutive et sont donc gourmands

L'utilisation dans l'alignement multiple dégrade considérablement la qualité et augmente le temps d'exécution.

2.4. K-means

Dans notre approche, nous avons proposé d'utiliser l'algorithme k-means pour le clustering des séquences biologiques. Selon notre connaissance, l'algorithme K-means n'a jamais été intégré dans une solution d'alignement multiple.

Le K-means (Figure 2.2) est un algorithme pour le partitionnement des données en générale, le problème traité est de diviser un ensemble de données en k groupe ; donc la variable k doit être définie au début. La fonction à minimiser est la somme des carrés des distances entre un centroide et l'ensemble de ses séquences.

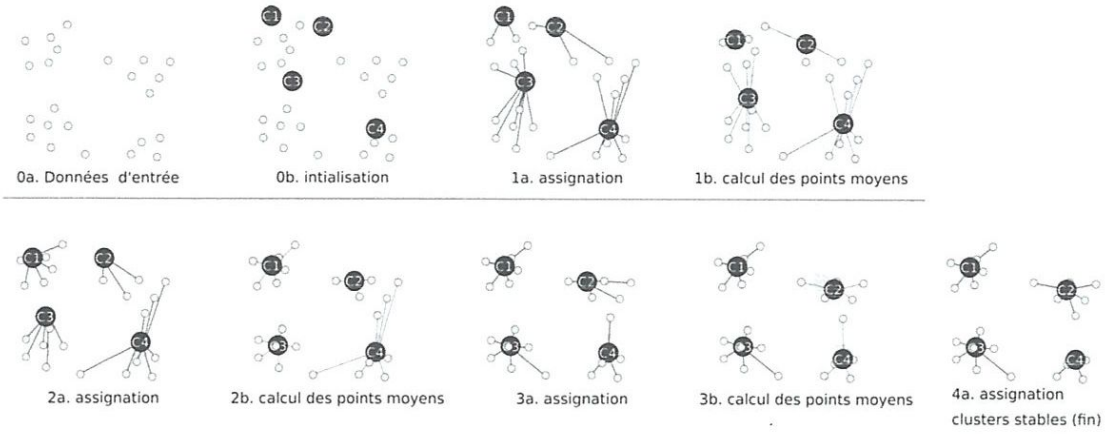

Figure 2. 2 : *L'algorithme K-means [19]*

Dans notre travail, on a proposé d'initialiser le k représentant le nombre de cluster en nombre de cœurs du processeur sur lequel va s'exécuter l'alignement. Donc, pour bénéficier des architectures multi cœurs, on a proposé l'alignement des clusters parallèlement afin de gagner en temps d'exécution. Les données à clusterer sont représentées par les séquences biologiques à aligner.

2.5. Conclusion :

Dans ce chapitre, on a présenté deux algorithmes les plus utilisés dans le clustering des séquences biologiques ; vu que ces algorithmes sont destinés à la détection des redondances, leurs utilisations dans le domaine de l'alignement des séquences peuvent engendrer des conséquences fatales. L'algorithme k-means pourra représenter une bonne alternative, vu qu'une variable k doit être initialisé avant le traitement. Dans notre cas elle représente le nombre de clusters de séquences à générer et au même temps, le nombre de cœurs de processeurs sur lequel l'alignement va s'exécuter. Ce qui engendre un algorithme très lié à l'architecture sur lequel il va s'exécuter.

3. Notre approche

3. 1. Introduction

Toutes les techniques utilisées pour résoudre le problème de MSA sont gourmandes en temps de calcul. Pour résoudre ce problème, deux classes de techniques ont été développées. La première est basée sur l'utilisation du parallélisme grâce à une approche matérielle. Certaines méthodes utilisent des ordinateurs à mémoire partagée et à mémoire distribuée, par ex. ClustalW-MPI [20] et Parallel T-Coffee[21]. Récemment, Church PC [22] a présenté une conception d'algorithmes d'alignement de séquences multiples sur des supercalculateurs à mémoire parallèle et distribuée. D'autre part, les unités de traitement graphique sont utilisées pour accélérer les programmes MSA, ex. GPU-Blast[23], G-MSA[24].

La deuxième classe consiste en l'utilisation du parallélisme par une approche logicielle en utilisant des modèles de programmation parallèles. Dans cette section, quatre techniques ont été proposés:

La première est une approche parallèle dans le calcul de la matrice de score. Elle consiste à calculer la matrice de score en parallèle dans le cas des parties non liées, Zafalon G [25] a montré que sa technique peut apporter une amélioration de 15% en temps d'exécution.

La seconde est une approche Pineline; Agarwal P [26] a proposé une technique avec un pipeline à deux étapes qui peut améliorer la complexité du problème. Ensuite, un nouveau pipeline multi-alignement pour les données de séquençage à haut débit est présenté par Shunping H[27].

Le troisième est une approche parallèle avec un algorithme dynamique. Ces techniques sont basées sur la parallélisation des algorithmes optimisés connus dans le domaine, par exemple, la parallélisation de l'algorithme Needleman & Wunsch par Naveed T[28], la parallélisation de l'algorithme Smith & Waterman de Dohi et al et la parallélisation de La technique optimale pour résoudre le MSA par Manal Helal et al sur l'architecture GPU.

Le quatrième est une approche parallèle des données ;. Cette technique a été proposée par Fahad S & al [29] dans laquelle ils ont proposé la technique k-mer pour faire le regroupement.

Plusieurs techniques de regroupement ont été proposées, telles que Uclust[30], CD-Hit[31], BlastClust[32], etc. Cela a permis de créer plusieurs approches pour MSA telles que Xiangyuan

Z & al [33] qui proposait également une approche parallèle basée sur les deux systèmes de clustering UClust et CD-HIT dans l'étape de cluster et MUSCL dans l'étape d'alignement.

Malgré le gain de temps d'exécution, ces techniques diminuent la qualité des alignements en raison des erreurs générées lors de la mise en cluster non hiérarchique.

Dans ce travail, nous proposons un algorithme de recherche locale basé sur l'algorithme K-means utilisant le parallélisme afin de décomposer les séquences en plusieurs clusters de bonne qualité sans perte de qualité d'alignement.

3. 2. Détail de l'approche

Notre approche comporte quatre étapes principales: regrouper des séquences en fonction du nombre de cœurs de processeur à l'aide de l'algorithme K-means, aligner chaque sous-ensemble et générer une séquence consensus pour chacun d'eux, et en fin, aligner toutes les séquences consensus générées en utilisant un algorithme d'alignement pour générer l'alignement Multiple. Le résumé des différentes étapes de notre approche est illustré à la .

Figure 3. 1 : *Résumé des différentes étapes de notre approche*

3.2.1. Clustring des sequence :

Les algorithmes de classification de séquence tentent de regrouper les séquences biologiques associées. Ils sont utilisés pour prédire l'homologie et la fonction, réduire la redondance, générer des sous-ensembles exploitables pour des méthodes plus gourmandes en temps d'exécution, comparer les données de différents environnements et quantifier la diversité des écosystèmes. En général, le clustering est appliqué de manière omniprésente. De nombreuses méthodes sont actuellement disponibles pour regrouper des séquences biologiques en familles et la plupart d'entre elles peuvent être classées en trois groupes principaux: les méthodes hiérarchiques, les méthodes basées sur les graphes et les méthodes de partitionnement. Plusieurs algorithmes ont été créés dans ce domaine, tels que: UClust [30], CD-HIT [31], BLASTClust [32], etc. Pour choisir un algorithme de classification, nous avons un ensemble général de fonctionnalités souhaitées telles que l'évolutivité et la sensibilité.

L'utilisation d'algorithmes de classification non hiérarchiques peut apporter un gain considérable en temps de calcul. Néanmoins, cela entraîne une perte importante de qualité de l'alignement.

Dans notre stratégie, nous avions proposé d'utiliser une technique basée sur l'algorithme de recherche locale k-means, créant un état initial contenant plusieurs clusters et l'améliorant de manière itérative jusqu'à l'obtention d'un bon Clustering.

Dans notre étude nous avons utilisé l'algorithme k-means .Il est constitué de plusieurs étapes (figure 3.2):

- Trier les séquences
- Choisir la première séquence
- Création des distance :
    - Calculer les scores de l'alignement entre la séquence choisie et toutes les autres en utilisant les algorithmes Needlman &wancsh pour l'alignement global et Smith & Waterman pour l'alignement local, et les mettre dans les deux premières colonnes d'un tableau
    - Rajouter une colonne indiquant la longueur de la séquence

- Faire un clustering de ce tableau en utilisant l'algorithme k-means avec k égale au nombre de cœurs de processeur sur lequel l'algorithme va s'exécuter ;

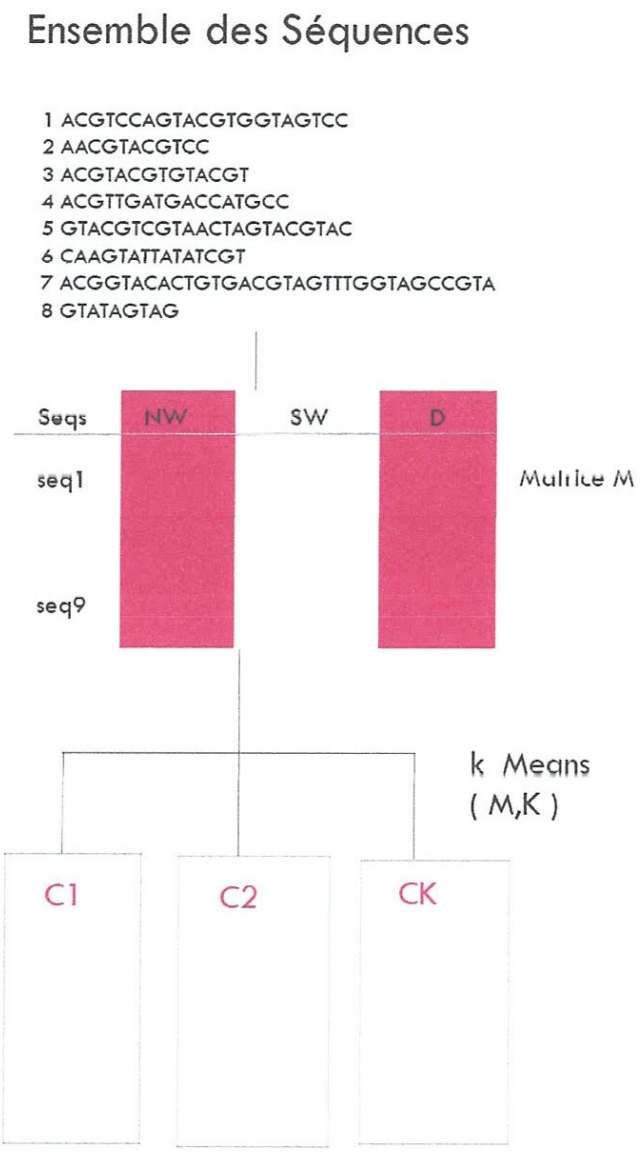

***Figure 3. 2** : Déroulement de l'étape de clustering avec l'algorithme k-means.*

3.2.2. Alignement des séquences et alignement des consensus

Nous avons utilisé les algorithmes MUSCLE, Clustal Omega et ClustalW pour aligner les séquences dans les clusters ainsi que dans la phase de consensus. La stratégie développée peut être considérée comme une étape préalable pour explorer un alignement prometteur avec tout autre aligneur.

3. 3. Temps d'exécution

Pour mesurer le gain fourni par notre approche qui utilise une étape de clustering avant l'alignement, nous avons effectué un ensemble de tests basé sur de grands ensembles de données contenant des ensembles de séquences de référence générés par GenRGenS [34]sur des profils de séquences réelles dérivées de BAliBASE dans lesquelles la longueur d'une séquence varie de 500 à 2000. Le nombre de séquences et leurs longueurs ont un effet direct sur le temps d'exécution de MSA.

L'approche proposée est implémentée dans MATLAB R2014b et nous avons effectué ces ensembles de tests sur un Intel Xeon ES2620 avec 6 cœurs, fonctionnant à 2,00 GHz, avec Caches (L1D- cache 32 Ko, L1I- cache 32Ko, L2- cache 256 Ko, L3-Cache 15 Mo) avec une mémoire DDR3 de 16 Go.

Le (tableau 3.1) montre la différence entre le temps d'exécution entre notre technique et les plus utilisés.

Nombre de sequence	Temps d'exécution			
	MUSCLE+ Kmeans	MUSCLE	Clustal Omega+ Kmeans	Clustal Omega
100	85	161	36	82
150	100	170	39	103
200	130	259	43	111
250	170	328	45	146
300	210	458	48	153
350	287	789	52	157
400	367	1046	58	187
450	502	1064	60	215

Tableau 3. 1 *: Comparaison du temps d'execution*

Le (Tableau 3.1) montre une grande différence de temps d'exécution si le Clustering est utilisé par rapport à sa non-utilisation.

Les performances sont également analysées en termes de temps d'exécution et d'évolutivité dans le cas de l'utilisation du parallélisme. Des expériences sont effectuées sur des jeux de données contenant les huit tests et MUSCLE en tant qu'aligneur.

Nombre de sequence	Temps d'exécution (s)		
	2 Cores	4 Cores	6 Cores
350	287	281	272
400	367	339	311
450	502	454	429
500	597	561	501
550	842	693	618
600	1058	910	824
650	1268	1112	1028
700	1630	1386	1140

Tableau 3. 2 *: temps d'exécution en utilisant le parallélisme*

3. 4. Qualité d'alignement

Pour étudier la qualité de l'alignement, nous comparons nos résultats avec ceux produits par les principales techniques d'alignement les plus utilisées.

La performance de notre approche a été testée sur la collection t BALIBASE v3. Nous avons comparé nos résultats avec d'autres algorithmes bien connus tels que CLUSTALW, Clustal Omega, Muscle

Pour mesurer la qualité de l'alignement, le programme qScore [35] est utilisé. Le programme affiche les scores suivants: Le score PREFAB Q (également connu sous le nom Balibase SPS score ou le Score du développeur) et le Ballbase TC (colonne totale).

La suite de benchmark BAliBASE contient des alignements de séquences multiples, organisés en 9 ensembles de référence représentant des problèmes spécifiques de MSA, y compris un petit nombre de séquences, des distributions phylogénétiques inégales, des extensions N / C-terminales ou des insertions internes, des répétitions, des domaines inversés et des régions transmembranaires [36].

	BB 11	BB 12	BB 20	BB 30	BB 40	BB 50
Clustal Omega	0,726 /0,523	0,911 / 0,857	0,646 /0,458	0,719 /0,504	0,901 /0,804	0,900 /0,808
Clustal Omega avec k-means	0,647 /0,427	0,889 /0,826	0,584 /0,399	0,677 /0,439	0,851 /0,745	0,816 /0,682
Clustal W Q/TC	0,667 /0,471	0,893 /0,831	0,612 /0,441	0,700 /0,475	0,863 /0,716	0,801 /0,628
Clustal W avec k-means	0,502 /0,305	0,833 /0,729	0,465 /0,262	0,573 /0,315	0,763 /0,593	0,754 /0,613
Muscle 4 Q/TC	0,744 /0,556	0,908 /0,853	0,735 /0,612	0,766 /0,582	0,858 /0,712	0,899 /0,790
Muscle Avec k-means	0,610 /0,369	0,877 /0,804	0,596 /0,417	0,667 /0,439	0,822 /0,674	0,823 /0,666

Tableau 3. 3 : *Comparaison des performances sur BAliBase 3.0*

3. 5. Discussions

Selon le tableau3.1, les algorithmes, Clustal W et Muscle échouent dans le cas de grandes données. Le temps de calcul pour les autres algorithmes (Notre solution, ClustalOmega) est acceptable.

En termes de qualité d'alignement et selon le *Tableau 3. 3 : Comparaison des performances sur BAliBase 3.0*

Tableau 2.1 : Comparaison des performances sur BAliBase 3.0 et après l'élimination des autres algorithmes qui échouent dans le cas des grandes données, et après avoir comparé l'application de notre approche avec les différents algorithmes d'alignement, on constate que malgré la petite perte de qualité d'alignement, notre solution reste toujours dans le même groupe que les autres algorithmes les plus utilisés dans le domaine néanmoins elle donne .

4. Implementation

4.1. Introduction

Dans ce chapitre, nous allons présenter l'implémentation de notre application d'alignement multiple de séquences. Notre application fait l'alignement multiple en proposant l'utilisation du clustering comme étape préalable selon l'approche proposée et calcule la qualité de l'alignement en utilisant un fichier de référence.

4.2. Objectif de notre application

L'objectif principale de notre projet est de construire une application qui permet d'aligner un ensemble de séquences biologique ADN ou protéine en essayant d'avoir un bon compromis Temps /qualité d'alignement. Les différents objectifs visés par un alignement multiple de séquences sont:

- Trouver les parties homologues
- Identification de résidus importants (conservés)
- Extraction de motifs communs
- Génération de séquences consensus

4.3. Présentation de langage de programmation

4.3.1. Matlab

Nous avons choisi l'environnement de développement Matlab R2014b qui est un langage de développement informatique particulièrement dédié aux applications scientifiques.

MATLAB est un logiciel de calcul matriciel à syntaxe simple. Avec ses fonctions spécialisées, MATLAB peut être aussi considéré comme un langage de programmation adapté pour les problèmes scientifiques, est utilisé pour développer des solutions nécessitant une très grande puissance de calcul.

Les graphiques intégrés permettent de visualiser facilement les données afin d'en dégager des informations. Grâce à sa bibliothèque et sa boîte à outils prédéfinie, ce qui encourage l'expérimentation, l'exploration et la découverte

4.3.2. Outils utilisés

Nous avons utilisés un d'outils qscore :

- **Qscore :** c'est un programme qui compare deux alignements de séquences multiples: un alignement à évaluer et un deuxième alignement qui est censé être correct (l'alignement "référence") [35].

Afin d'évaluer notre solution, on l'a comparé avec les solutions des algorithmes les plus utilisés, donc, dans notre application, on a intégré les algorithmes suivants : Cluslatlw, ClustalO, Muscle, Maftt.

4.3.3. Format fasta

Une séquence au format FASTA commence par une ligne de titre (nom, définition ...), suivie par les lignes de la séquence. La ligne de titre se distingue de la séquence par un symbole plus grand que (">") en début de ligne. La longueur de cette ligne ne doit pas excéder 200 caractères. Il est recommandé de mettre la séquence sous forme de lignes de 80 caractères maximum. Un exemple de séquence au format fasta (Figure 4.1)

```
>gi|532319|pir|TVFV2E|TVFV2E envelope protein
ELRLRYCAPAGFALLKCNDADYDGFKTNCSNVSVVHCTNLMNTTVTTGLLLNGSYSENRT
QIWQKHRTSNDSALILLNKHYNLTVTCKRPGNKTVLPVTIMAGLVFHSQKYNLRLRQAWC
HFPSNWKGAWKEVKEEIVNLPKERYRGTNDPKRIFFQRQWGDPETANLWFNCHGEFFYCK
MDWFLNYLNNLTVDADHNECKNTSGTKSGNKRAPGPCVQRTYVACHIRSVIIWLETISKK
TYAPPREGHLECTSTVTGMTVELNYIPKNRTNVTLSPQIESIWAAELDRYKLVEITPIGF
APTEVRRYTGGHERQKRVPFVXXXXXXXXXXXXXXXXXXXXXXVQSQHLLAGILQQQKNL
LAAVEAQQQMLKLTIWGVK
```

***Figure 4. 1** : Format fasta*

C'est l'un des standards les plus utilisés en bioinformatique et la plupart des logiciels reconnaissent ce format.

4.4. Interface

L'interface graphique est divisée en deux parties (*Figure 4.2*): la première partie Align Multiple Séquences; pour aligner des séquences dans des fichiers FASTA, et la deuxième pour comparer MSA avec des références, pour comparer un ou plusieurs fichiers contenant des alignements avec des fichiers contenant des références. La figure suivante (Figure 4.2) présente l'interface principale de notre application.

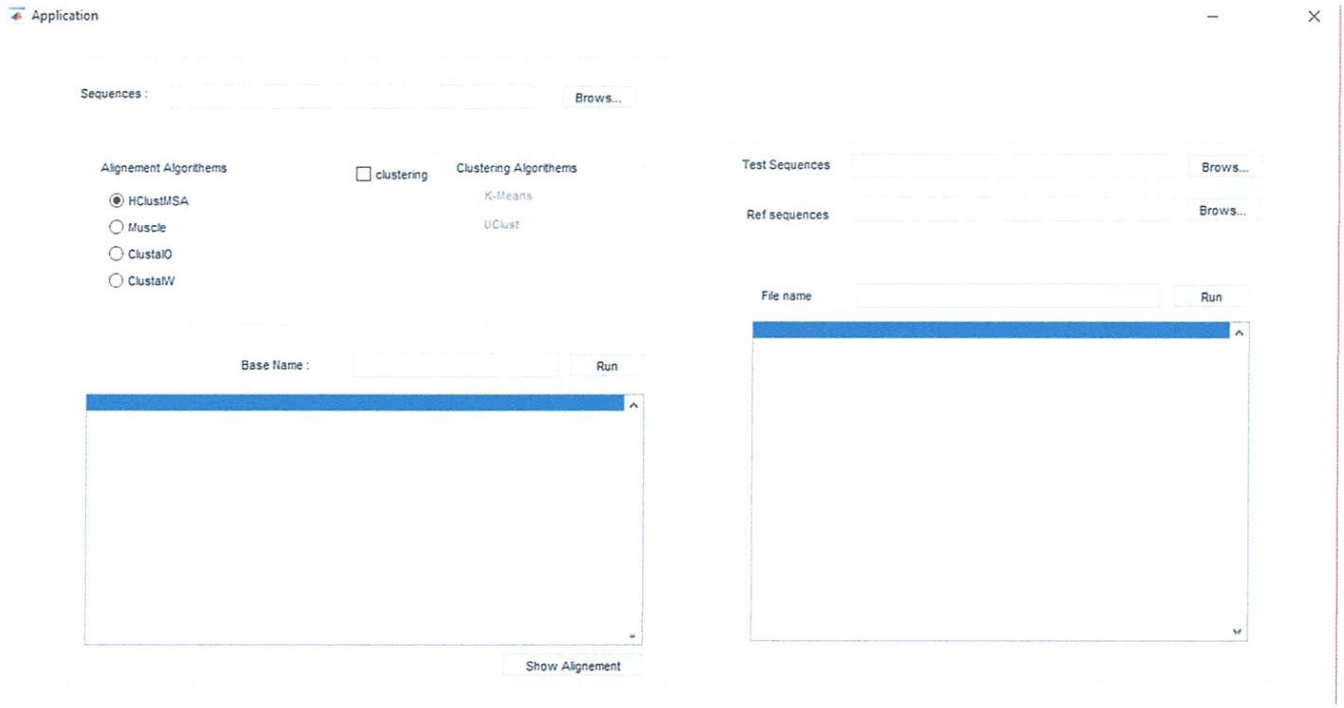

***Figure 4. 2** :Interface Principale.*

4.4.1. L'alignement multiple de séquences

Avant de faire un MSA, l'utilisateur doit choisir un algorithme d'alignement. L'utilisation du Clustering est optionnelle et l'algorithme k-means est utilisé.

Lorsqu'on clique sur le bouton Brows, on peut parcourir les bases des séquences et choisir le fichier FASTA contenant les séquences à aligner. Les figures suivantes (Figure 4.3) affichent un exemple des séquences choisies par l'utilisateur.

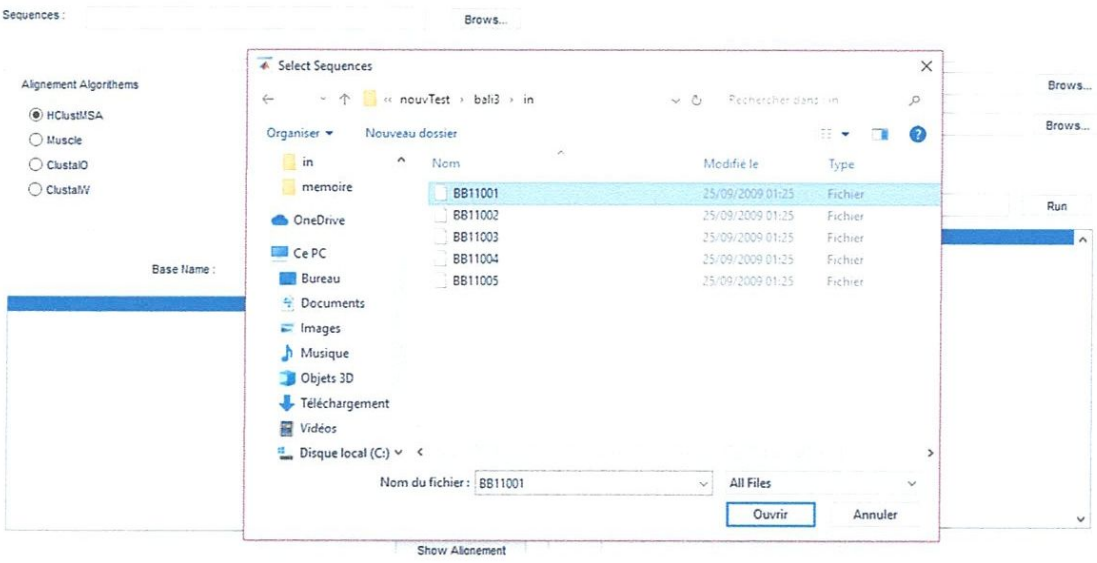

***Figure 4. 3** : Sélection du dossier des séquences*

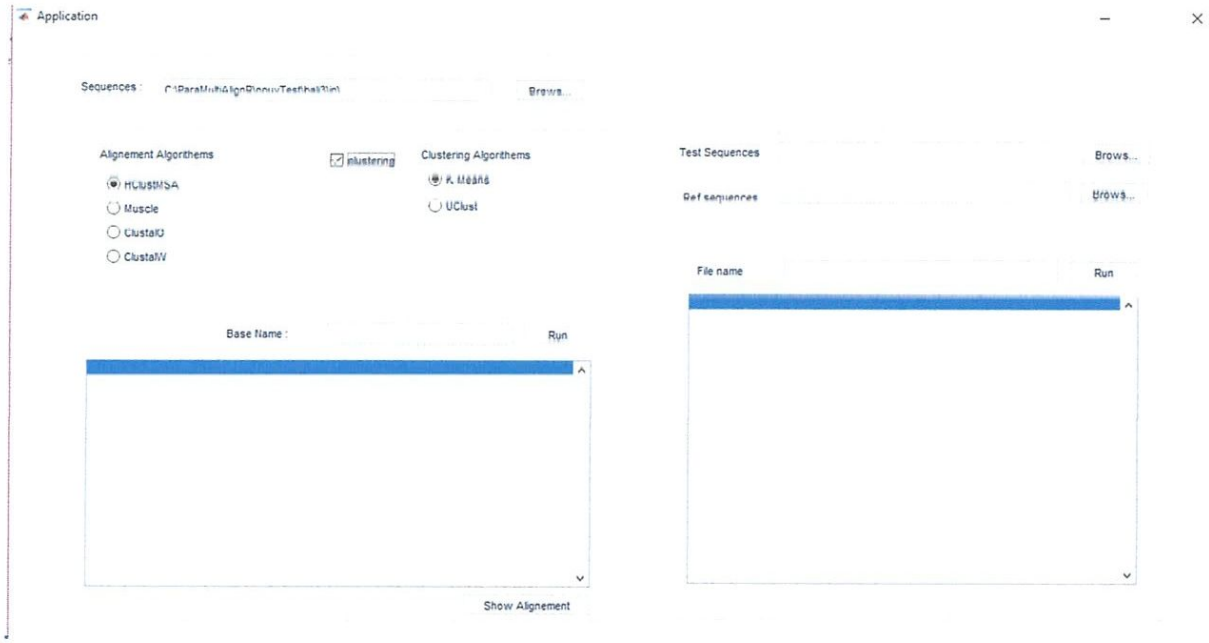

***Figure 4. 4 :** Choix d'un algorithme de Clustring .*

Avant de cliquer sur le bouton Run pour lancer l'alignement il faut écrire le nom de base (Figure 4.5,Figure 4.6).

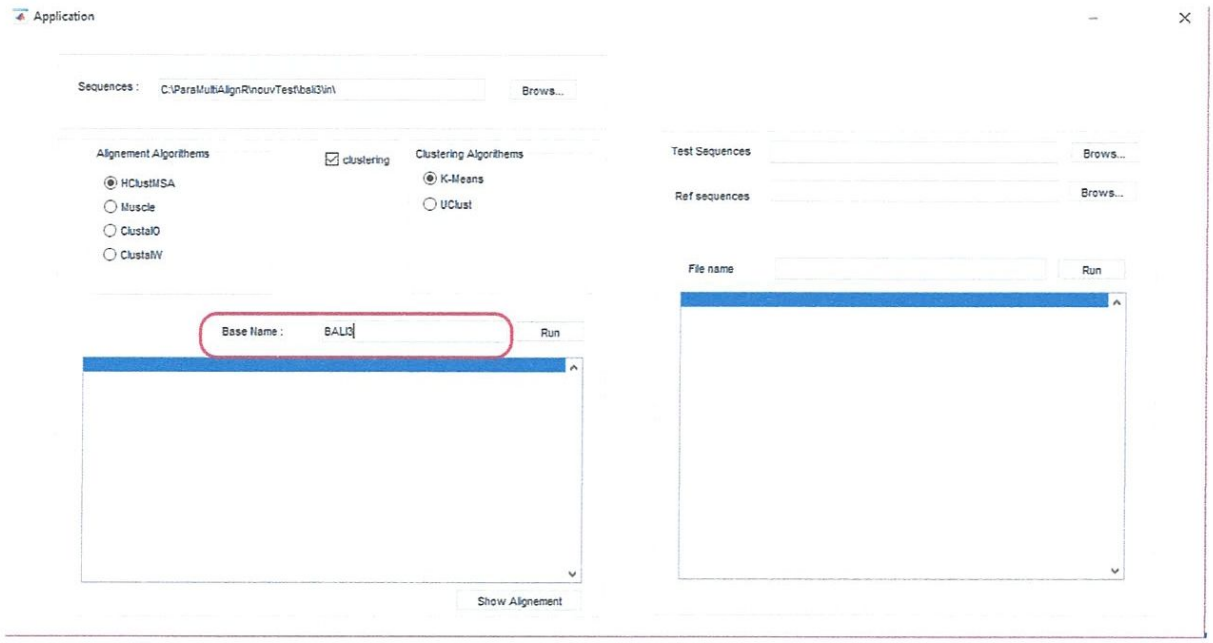

Figure 4. 5 : *Nommer les résultats*

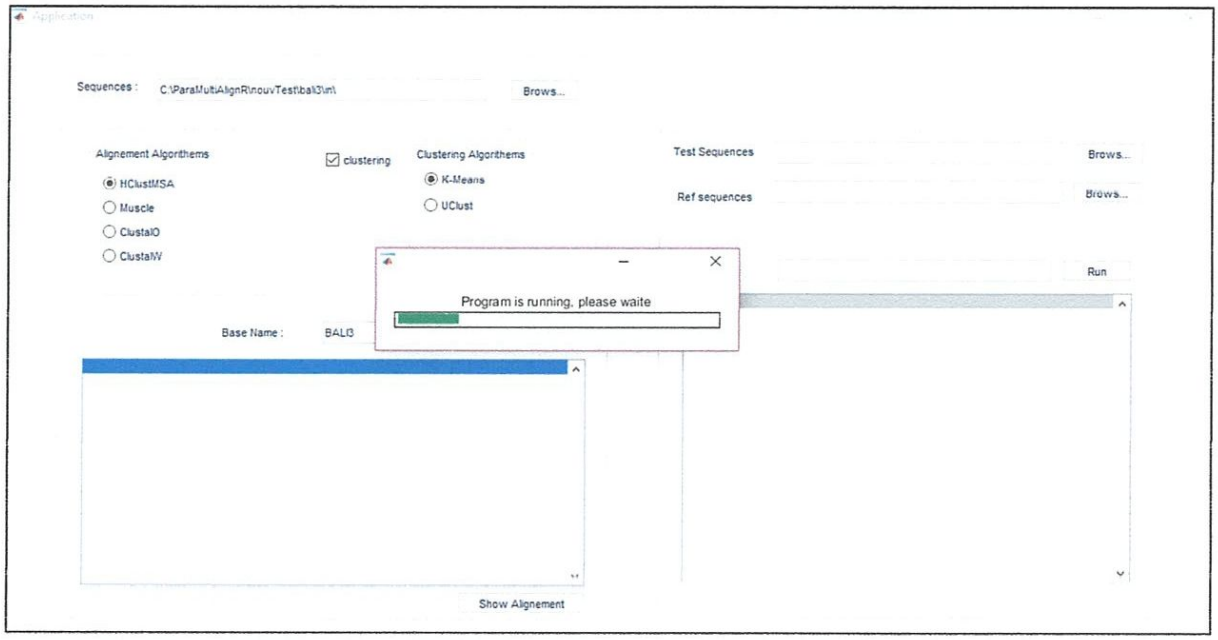

Figure 4. 6 : *Nommer les résultats*

Une fois l'alignement est terminé le temps de l'alignement multiple de chaque fichier est affiché et enregistré dans un fichier csv (time.csv) (Figure 4.7, Figure 4.8).

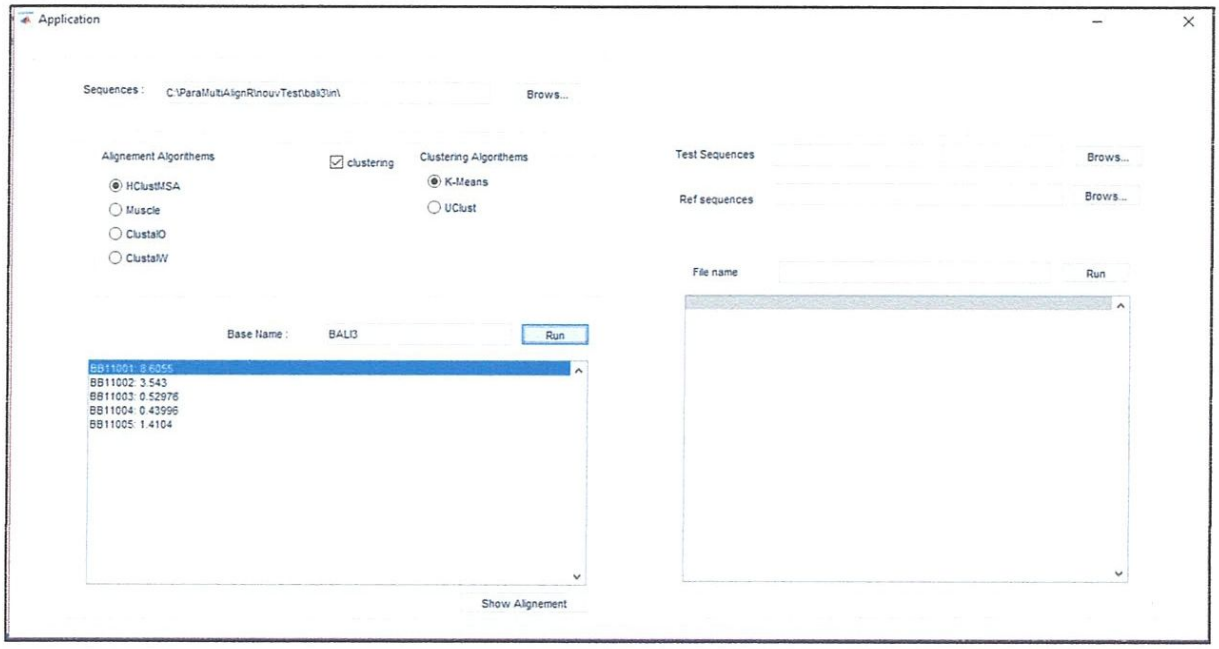

Figure 4. 7 : *Fin de l'alignement.*

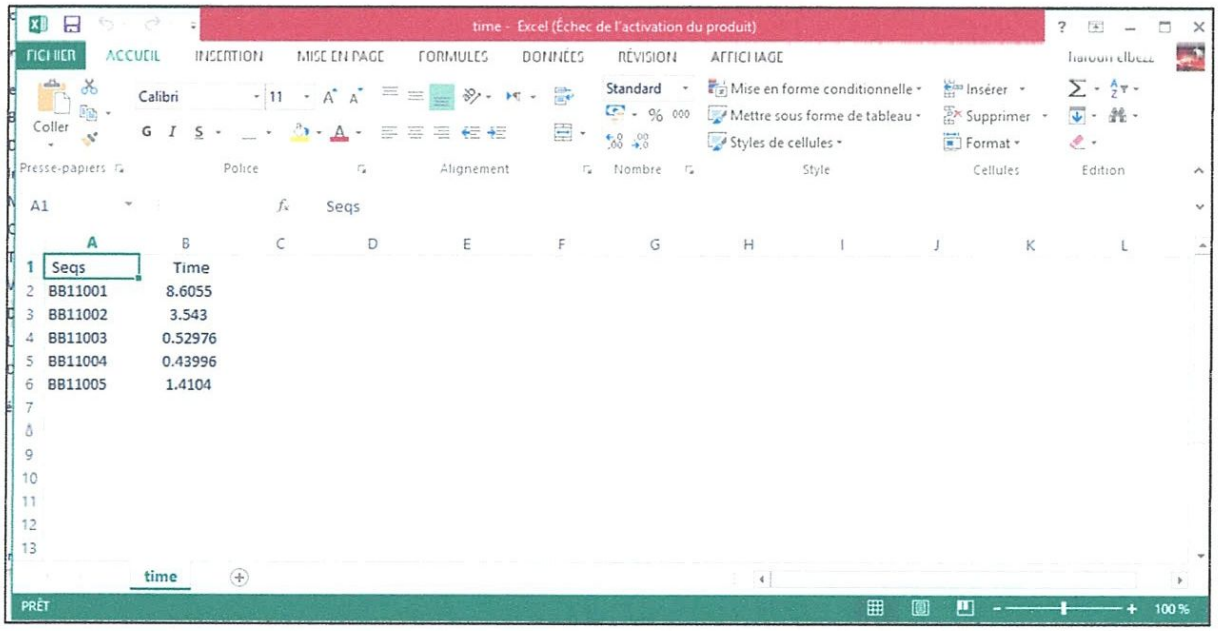

Figure 4. 8 *: Fichier CSV contenant les temps d'execution.*

Pour afficher les séquences alignées, on choisit le fichier contenant les séquences déjà alignées et on clique sur le bouton Show Alignement (*Figure 4.9*).

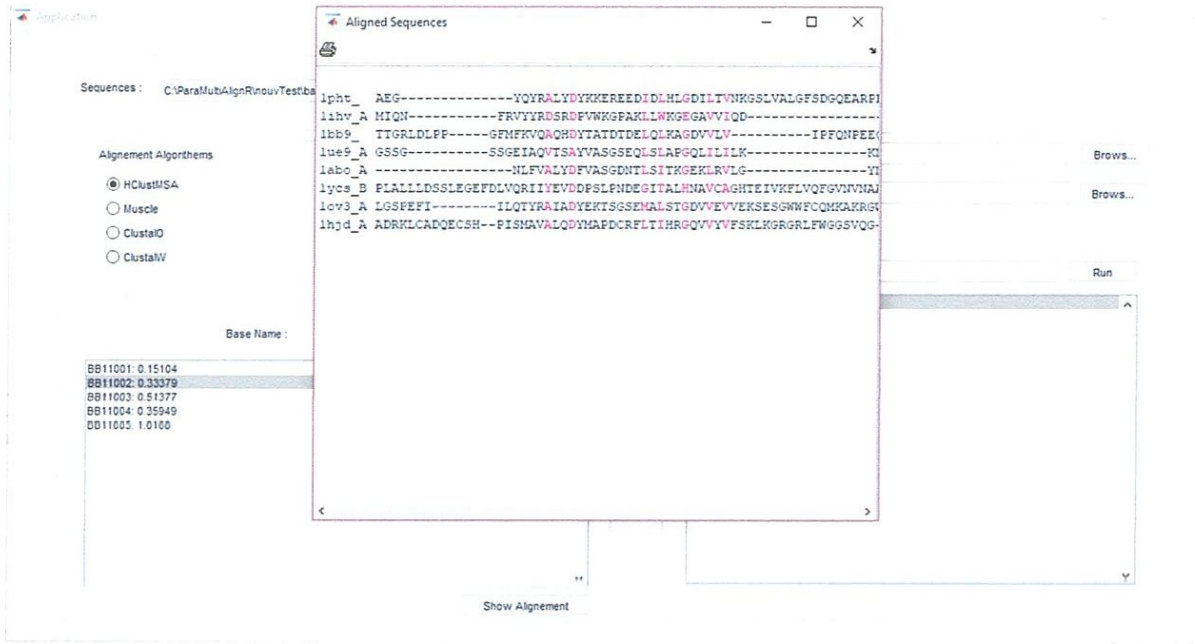

Figure 4. 9 *: Alignement multiple des séquences*

4.4.2. Mesurer la qualité d'alignement multiple de séquences

Le bouton Brows, permet de parcourir les bases des séquences à mesurer qui doivent être alignées puis on choisit les séquences de références et ainsi calculer la qualité d'alignement en utilisant l'application QScore déjà présentée. Les figures (Figure 4.10) et (Figure 4.11) et (Figure 4.12*)* montrent un exemple en choisissant un fichier de séquences (Fasta).

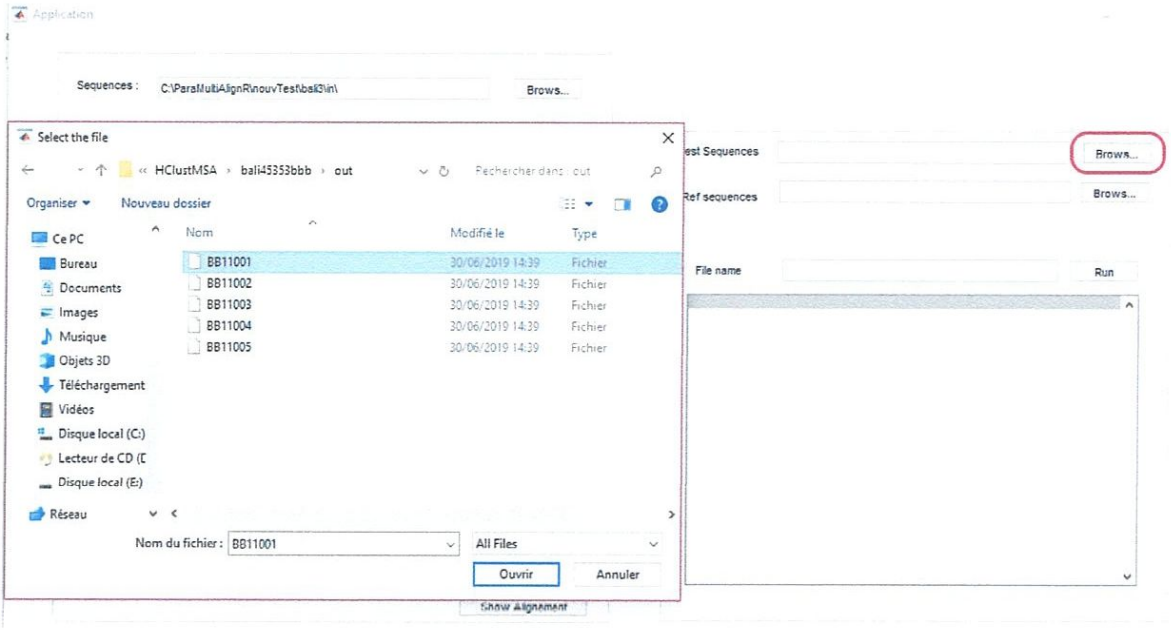

Figure 4. 10 . Dossier contenant les séquences à tester

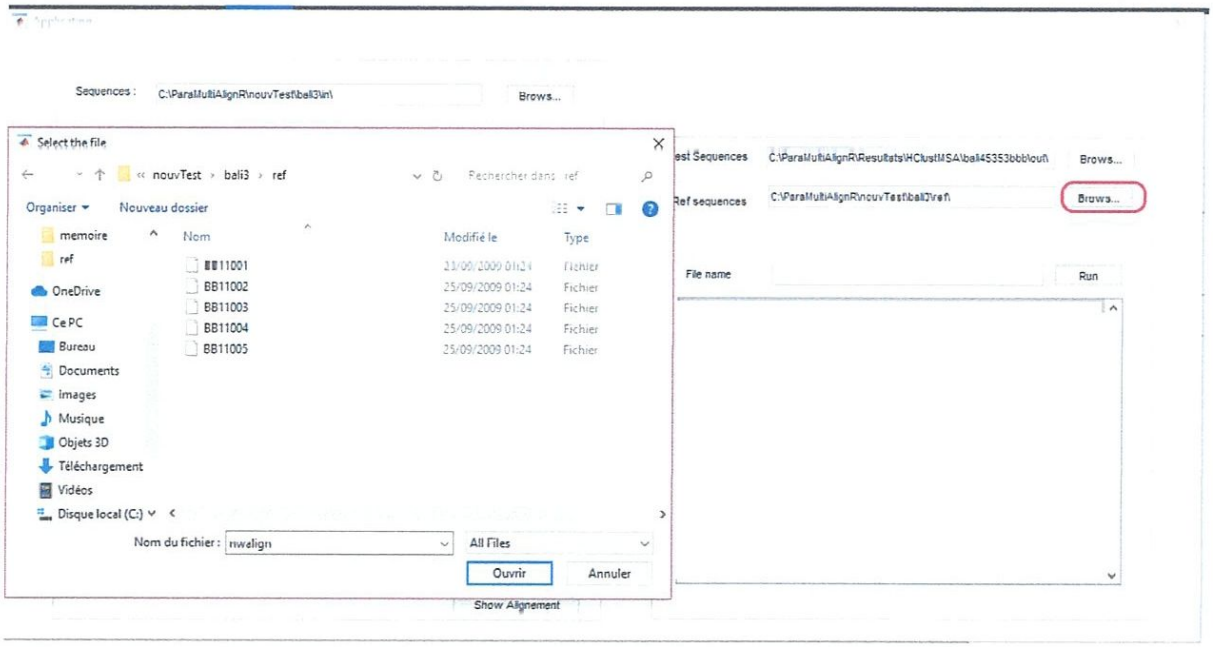

***Figure 4. 11** : Dossier contenant les séquences de référence.*

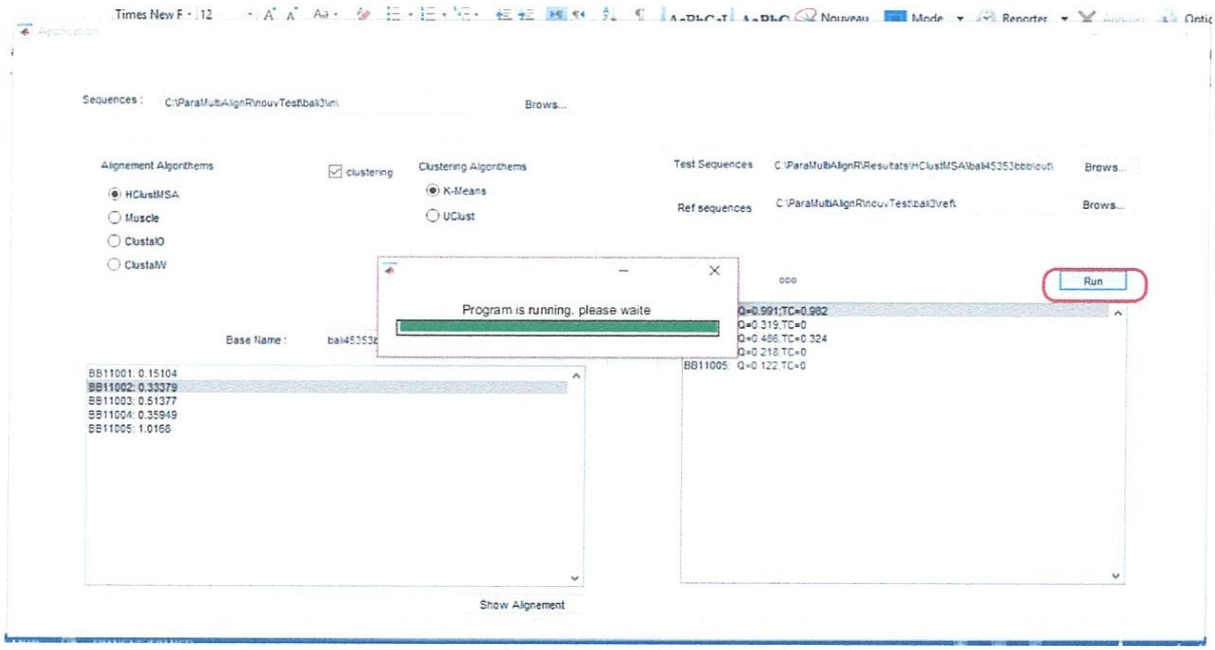

***Figure 4. 12** : Calcule de la qualité*

A la fin des calculs, les résultats seront affichés et enregistrés dans un fichier csv, (*Figure 4.13, Figure 4.14).*

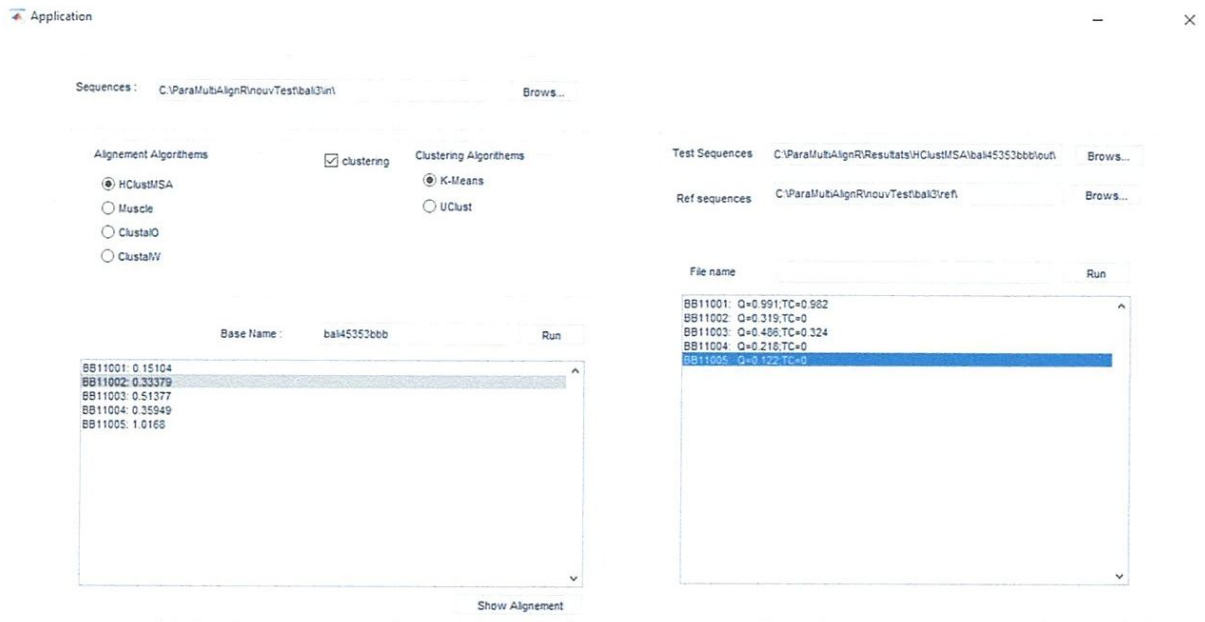

***Figure 4. 13 :** Qualités des alignements*

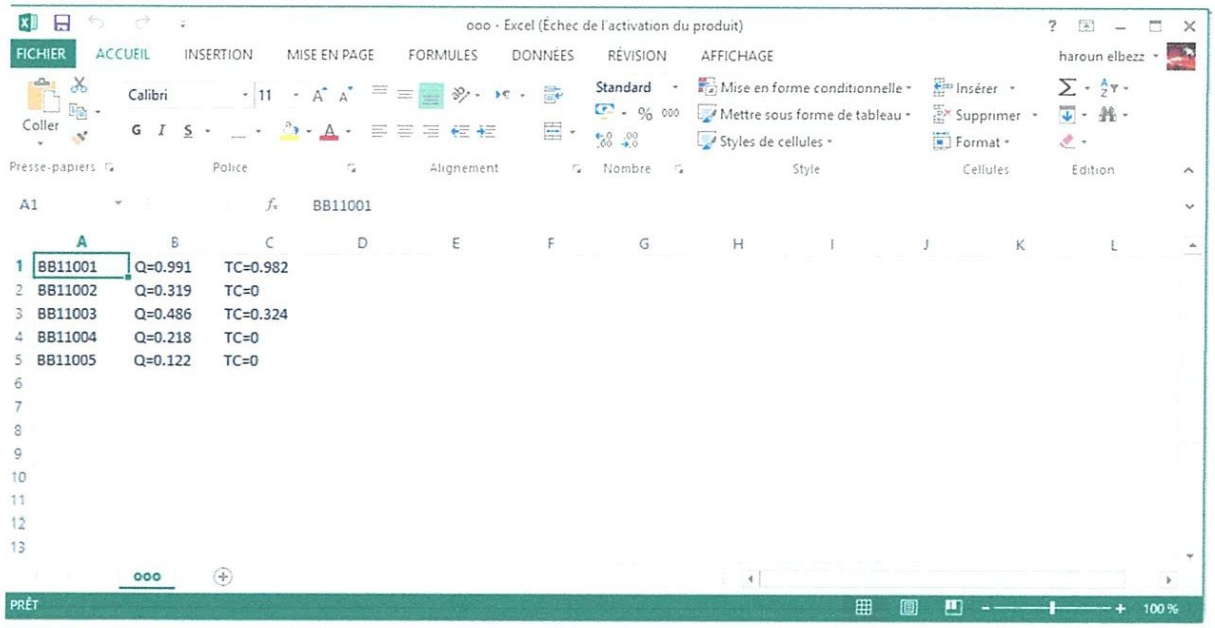

***Figure 4. 14** : Qualités des alignements.*

4.5. Conclusion

Dans ce chapitre nous avons présenté notre outil d'alignement de séquences biologiques. Notre outil peut aligner des séquences biologiques en utilisant en choix plusieurs algorithmes.

La nouveauté, consiste en l'utilisation d'une étape de préparation des données améliorant ainsi, considérablement le temps d'exécution, c'est bien le Clustering basé sur l'utilisation de l'algorithme k-means.

Comme deuxième fonctionnalité, notre outil donne la possibilité de faire des comparaisons entre les différents algorithmes d'alignement en faisant des comparaisons avec les séquences de référence.

Conclusion générale

Dans ce travail, nous avons présenté une nouvelle technique d'optimisation pour l'alignement des séquences multiples en bioinformatique. Dans notre technique, l'alignement implique deux étapes principales: le premier est le regroupement de séquences en sous-ensembles, et cela a une grande amélioration dans le temps d'exécution dans le cas de séries de données à grande échelle, et permet également l'utilisation du parallélisme dans le cas des ordinateurs multi-cœur, ce qui a permis d'éviter l'échec apparut dans la plus part des méthodes MSA pour aligner un grand nombre de séquences. Dans cette étape, on a proposé l'utilisation de l'algorithme k-mean avec les deux algorithmes de la programmation dynamique, le Needlman & Wanch pour une comparason globale et Smith & waterman pour une comparaison locale.

La grande caractéristique de notre approche est sa simplicité et sa capacité à fournir une plateforme extensible pour améliorer d'autres programmes d'alignement.

Notre travail a abouti à la création d'un outil d'alignement de séquence utilisant différents algorithmes qui pourra être utile aussi bien pour biologiste pour l'alignement de leurs séquences ainsi que pour les développeurs des algorithmes d'alignement afin de comparer leurs résultats.

Références

[1] WANG, Lusheng et JIANG, *On the complexity of multiple sequence alignment.* Journal of computational biology, 1994. **1**(4): p. 337-348.

[2] NOTREDAME, Cédric, *Recent progress in multiple sequence alignment: a survey.* Pharmacogenomics, 2002. **3**(1): p. 131-144.

[3] NEEDLEMAN, Saul B. WUNSCH, Christian D ,*A general method applicable to the search for similarities in the amino acid sequence of two proteins.* Journal of molecular biology, 1970. **48**(3): p. 443-453.

[4] FENG, Da-Fei DOOLITTLE, Russell F, *Progressive sequence alignment as a prerequisitetto correct phylogenetic trees.* Journal of molecular evolution, 1987. **4**(54): p. 351-360.

[5] Layeb, A., *Approche quantique évolutionnaire pour l'alignement multiple de séquences en bioinformatique.* 2005.

[6] Hogeweg, Paulien, and Ben Hesper, *The alignment of sets of sequences and the construction of phyletic trees: an integrated method.* Journal of molecular evolution, 1984. **20**(2): p. 175-186.

[7] FENG, Da-Fei DOOLITTLE, Russell F, *Progressive sequence alignment as a prerequisitetto correct phylogenetic trees.* Journal of molecular evolution, 1987. **25**(4): p. 351-360.

[8] THOMPSON, Julie D., HIGGINS, Desmond G., et GIBSON, Toby J, *CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice.* Nucleic acids research, 1994. **22**(22): p. 4673-4680.

[9] Thompson, J.D., F. Plewniak, and O. Poch, *BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs.* Bioinformatics (Oxford, England), 1999. **15**(1): p. 87-88.

[10] Zola, J., et al., *PARALLEL-TCOFFEE: A parallel multiple sequence aligner.* ISCA PDCS, 2007. **7**: p. 248-253.

[11] Edgar, R.C., *MUSCLE: multiple sequence alignment with high accuracy and high throughput.* Nucleic acids research, 2004. **32**(5): p. 1792-1797.

[12] Morgenstern, B., et al., *DIALIGN: finding local similarities by multiple sequence alignment.* Bioinformatics (Oxford, England), 1998. **14**(3): p. 290-294.

[13] Thompson, J. D., Plewniak, F., & Poch, O, "BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs," *Bioinformatics,* vol. 15, no. 1, pp. 87-88, 1999..

[14] Edgar, R. C, "MUSCLE: multiple sequence alignment with high accuracy and high throughput," *Nucleic acids research,* vol. 32, no. 5, pp. 1792-1797, 2004..

[15] Edgar, R. C. (2010). "Search and clustering orders of magnitude faster than BLAST bioinformatics. 26 (19): 2460–2461. doi:10.1093/bioinformatics/btq461

[16] https://en.wikipedia.org/wiki/UCLUST#cite_note-Edgar2010-1 .

[17] https://drive5.com/usearch/manual/uclust_algo.html

[18] http://weizhongli-lab.org/cd-hit/

[19] https://fr.wikipedia.org/wiki/K-moyennes#/media/Fichier:K-means.png

[20] LI, Kuo-Bin, *ClustalW-MPI. ClustalW analysis using distributed and parallel computing. Bioinformatics.* 2003. **19**(12): p. 1585-1586.

[21] ZOLA, Jaroslaw, YANG, Xiao, ROSPONDEK, Adrian, *et al ,PARALLEL-TCOFFEE: A parallel multiple sequence aligner. ISCA PDCS.* 2007. **7**: p. 248-253.

[22] CHURCH, Philip C., GOSCINSKI, Andrzej, HOLT, Kathryn, *et al ,Design of multiple sequence alignment algorithms on parallel, distributed memory supercomputers in Engineering in Medicine and Biology Society, EMBC.* in *Annual International Conference of the IEEE.* 2011.

[23] VOUZIS, Panagiotis D. et SAHINIDIS, Nikolaos V, *GPU-BLAST: using graphics processors to accelerate protein sequence alignment. Bioinformatics.* 2011. **27**(2): p. 182-188.

[24] BLAZEWICZ, Jacek, FROHMBERG, Wojciech, KIERZYNKA, Michal, *et al, G-MSA—A GPU-based, fast and accurate algorithm for multiple sequence alignment.* Journal of Parallel and Distributed Computing, 2013. **73**(1): p. 32-41.

[25] ZAFALON, Geraldo FD, MARUCCI, Evandro A., MOMENTE, Julio C., *et al , Improvements in the score matrix calculation method using parallel score estimating algorithm.* Journal of biophysical chemistry, 2013, p. 47-51.

[26] AGARWAL, Pankaj et RIZVI, S. A. M, *Solving sequence alignment problem using pipeline approach.* Bharati Vidyapeeth's Institute of Computer Applications and Management, 2009: p. 107.

[27] HUANG, Shunping, HOLT, James, KAO, Chia-Yu, *et al, A novel multi-alignment pipeline for high-throughput sequencing data.* Database, 2014. **2014**: p. bau057.

[28] NAVEED, Tahir, SIDDIQUI, Imitaz Saeed, et AHMED, Shaftab ,*Parallel needleman-wunsch algorithm for grid. in Proceedings of the PAK-US International Symposium on High Capacity Optical Networks and Enabling Technologies (HONET 2005).* Islamabad, 2005.

[29] SAEED, Fahad et KHOKHAR, Ashfaq , *A domain decomposition strategy for alignment of multiple biological sequences on multiprocessor platforms.* Journal of Parallel and Distributed Computing, 2009. **69**(7): p. 666-677.

[30] EDGAR, Robert C, *Search and clustering orders of magnitude faster than BLAST.* Bioinformatics, 2010. **26**(19): p. 2460-2461.

[31] LI, Weizhong , *fast program for clustering and comparing large sets of protein or nucleotide sequences.* Bioinformatics, 2006. **22**(13): p. 1658-1659.

[32] DONDOSHANSKY, I. et WOLF, Y, *Blastclust (ncbi software development toolkit). NCBI, Bethesda, Md.* 2002.

[33] ZHU, Xiangyuan, LI, Kenli, et SALAH, Ahmad, *A data parallel strategy for aligning multiple biological sequences on multi-core computers.* Computers in biology and medicine, 2013. **43**(4): p. 350-361.

[34] Ponty, Y., M. Termier, and A. Denise, "GenRGenS: software for generating random genomic sequences and structures," *Bioinformatics,* vol. 22, no. 12, pp. 1534-1535, 2006.

[35] Edgar, R.C., "A quality scoring program," 2017. [Online]. Available: http://www.drive5.com/qscore/.

[36] Thompson, J.D., et al., "A comprehensive benchmark study of multiple sequence alignment methods: current challenges and future perspectives," *PloS one,* vol. 6, no. 3, p. e18093, 2011.