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 : STIC

Thème :

K-means & K-mers pour le regroupement et la comparaison de grands ensembles de séquences biologiques

Encadré Par :

Lebsir Rabeh

Présenté par :

Bousmat Yacine

Juin 2022

Dédicace

“

Je dédie ce modeste travail à Mr Lebsir Rabeh , mon tuteur de projet de fin d'étude, qui m'a suivi tout au long de cette période. A mes très chers parents
Mes très chers parents,
Mes chères sœurs,
Mon frère et sa femme,
Toute ma famille,
Tous mes amis.
Merci.

”

- Yacine

Remerciements

“

Je tiens tout d'abord à remercier Dieu le tout puissant, qui nous a donné la force et la patience d'accomplir ce modeste travail.
Mes chaleureux remerciements vont à ma famille ; « je vous présente mes années d'étude dans ce modeste travail juste pour voir les étincelles de fierté dans vos yeux ».
En second lieu, nous tenons à remercier notre encadreur Lebsir Rabeh, pour son précieux conseil et son aide durant toute la période du travail.
Mon vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre projet en acceptant d'examiner mon travail et de l'enrichir par leurs propositions.
Enfin, je tiens également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Bousmat Yacine
Juine 2022

”

Résumé

La bioinformatique est très importante pour extraire le plus d'informations des données biologiques. Les méthodes anciennes sont utiles, mais elles deviennent incapables à mesurer la quantité de données biologiques provenant de projets de séquençage à haut débit qui augmente sans cesse. L'un des domaines les plus importants de la bioinformatique est le regroupement de séquences. Dans ce manuscrit, nous nous concentrons sur le regroupement de séquences pour aider les algorithmes d'alignement de séquences multiples dans le cas de séquences biologiques à grande échelle avec une demande croissante en biologie computationnelle. Nous présentons notre méthode de clustering basé sur l'algorithme K-means guidé par les k-mers liées aux séquences à aligner. Nous intégrons aussi notre méthode de clustering dans une stratégie d'alignement multiple [1], afin de gagner en temps d'exécution sans perdre en qualité. Nous avons testé l'approche sur un processeur multi cœur avec un ensemble de Benchmarks connus dans la littérature. Nous avons comparé nos résultats avec ceux généré par l'algorithme de clustering UClust. Les résultats montrent que notre approche perd en termes de temps de calcul par rapport à UClust, tout en gardant de la précision dans tous les Benchmarks testés.

Mots clés : Séquence biologique, Alignement multiple de séquences, Clustering, Recherche Locale, Apprentissage automatique, Métaheuristique.

Table des matières

Table des figures

Liste des tableaux

List of Algorithms

Introduction générale

Contexte & Problématique

Il est devenu indispensable de développer des outils performants afin d'extraire un maximum d'informations pour bien comprendre le monde vivant. Bien que les anciens algorithmes bioinformatiques soient encore utilisables, ils deviennent de plus en plus incapables de traiter les masses importantes de données. Les domaines les plus importants de la bioinformatique comprennent le regroupement de séquences et l'alignement de séquences. L'alignement de séquences biologiques est une opération fondamentale en bioinformatique visant à identifier des régions conservées entre deux séquences. On suppose que des séquences similaires peuvent avoir les mêmes propriétés (physicochimiques ou structurelles).

Aligner deux séquences signifie écrire les deux séquences l'une au-dessus de l'autre pour extraire le plus de similarité possible. Si le problème d'alignement de deux séquences a été résolu exactement par les deux algorithmes de programmation dynamique, Needleman-Wunsch pour l'alignement global et Smith-Waterman pour l'alignement local, le problème de l'alignement multiple est encore beaucoup plus compliqué. En pratique, les alignements multiples peuvent examiner un ensemble de séquences pour extraire des modèles communs censés jouer un rôle dans la fonction ou la structure des séquences de la même famille. Il a été montré que les problèmes d'alignement multiple sont NP-complets et appartiennent donc à une classe de problèmes insolubles.
Par conséquent, les méthodes de résolution du problème d'alignement multiple peuvent être divisées en trois catégories principales :

- Les algorithmes exacts.
- Les algorithmes progressifs.
- Les algorithmes itératifs.

Motivations et contributions

Les techniques utilisées pour résoudre le problème MSA sont toutes couteuses en calcul, et la majorité d'entre elles échouent lorsqu'elles sont appliquées au Big Data. Dans cet esprit, nous avons présenté une stratégie pour l'alignement de séquences multiples basées sur le paradigme diviser pour régner afin de réduire le temps de calcul tout en garantissant de bons résultats. Elle est construite autour de quatre étapes clés : regrouper les séquences, aligner chaque sous-ensemble et produire un consensus, aligner tous les consensus créés sur des cœurs de processeurs distincts et, finalement, générer l'alignement complet [1].

Nous avons développé et utilisé une méthode de clustering basé sur l'algorithme k-means utilisant les k-mers, afin de faire une parallélisation des données et gagner ainsi en temps d'éxecution sans perdre en qualité d'alignement.

Nous avons testé notre technique avec une collection de Benchmarks connus dans la littérature (BALIBASE v3) et nous avons comparé nos résultats avec l'algorithme de clustering rapide UClust en utilisant ClustalW comme algorithme d'alignement. Pour mesurer le gain en temps de calcul fourni par notre approche qui utilise une étape de clustering avec le k-means, nous avons fait un ensemble de tests basé sur de grands jeux de données contenant des ensembles de séquences générés sur des profils de séquences réelles dérivés de BAliBASE. Les tests montrent que notre approche perd en temps d'exécution par rapport à UClust tout en gardant une qualité très proche à l'utilisation native de ClustalW.

Organisation du mémoire

En plus de l'introduction générale et la conclusion générale, le manuscrit est décomposé en deux parties chacune contient deux chapitres.

Le premier chapitre aborde les principaux concepts abordés afin de bien appréhender les concepts fondamentaux, notamment la biologie moléculaire et les principes de la bio-informatique, ainsi que le problème de l'alignement des séquences biologiques. Les idées fondamentales de l'intelligence computationnelle, les méthodes de calcul utilisées pour résoudre le problème de l'alignement de séquences multiples sont toutes abordées dans le chapitre 2.

La deuxième section est consacrée à la présentation de la contribution ; dans ce chapitre, nous passons en revue l'algorithme en détail, y compris ses avantages et ses inconvénients.

partie I

Etat de l'art

Chapitre 1

Bioinformatique

1.1 Introduction

L'étude des séquences biomoléculaires fait l'objet de la bioinformatique, domaine entre l'informatique et la biologie. Dans les années 1960, la première base de données biologiques a été créée et la séquence de 51 résidus de l'insuline bovine a été proclamée première séquence protéique. La première séquence d'acide nucléique, un ARNt d'alanine de levure de 77 bases, a été rapportée dix ans plus tard. Dayhoff a rassemblé toutes les données de séquençage disponibles et a produit la première base de données bioinformatique un an plus tard.

Dans ce chapitre, nous allons présenter les principaux concepts de la biologie moléculaire, la bioinformatique ainsi que l'alignement des séquences.

1.2 Biologie moléculaire

La biologie moléculaire est l'étude de la fonction des organismes vivants au niveau moléculaire, y compris la structure, la synthèse et la modification du matériel génétique (ADN, ARN) (mutation). Elle s'intéresse à la génétique, à la physique, à la biochimie et à l'informatique.

1.2.1 ADN

James Watson, un biologiste américain, et Francis Crick, un physicien britannique, ont fait la démonstration de leur célèbre modèle de double hélice ADN au début des années 1950. Leur proposition, basée sur l'analyse de modèles de diffraction des rayons X et le développement minutieux de modèles, s'est avérée valable et ouvre la voie à une meilleure compréhension de la fonction d'équipement génétique de l'ADN.

L'information génétique d'un organisme est stockée dans l'acide désoxyribonucléique (ADN). L'ADN est composé de deux brins qui sont torsadés ensemble pour former une double hélice. Chaque chaîne est composée de nucléotides, qui sont constitués d'un groupement phosphate, du désoxyribose, d'un sucre à cinq carbones (pentose) et de l'une des quatre bases azotées : adénine (A), guanine (G), thymine (T), et la cytosine (C) (C). La figure 1.1 [1] illustre la structure de l'ADN [2].

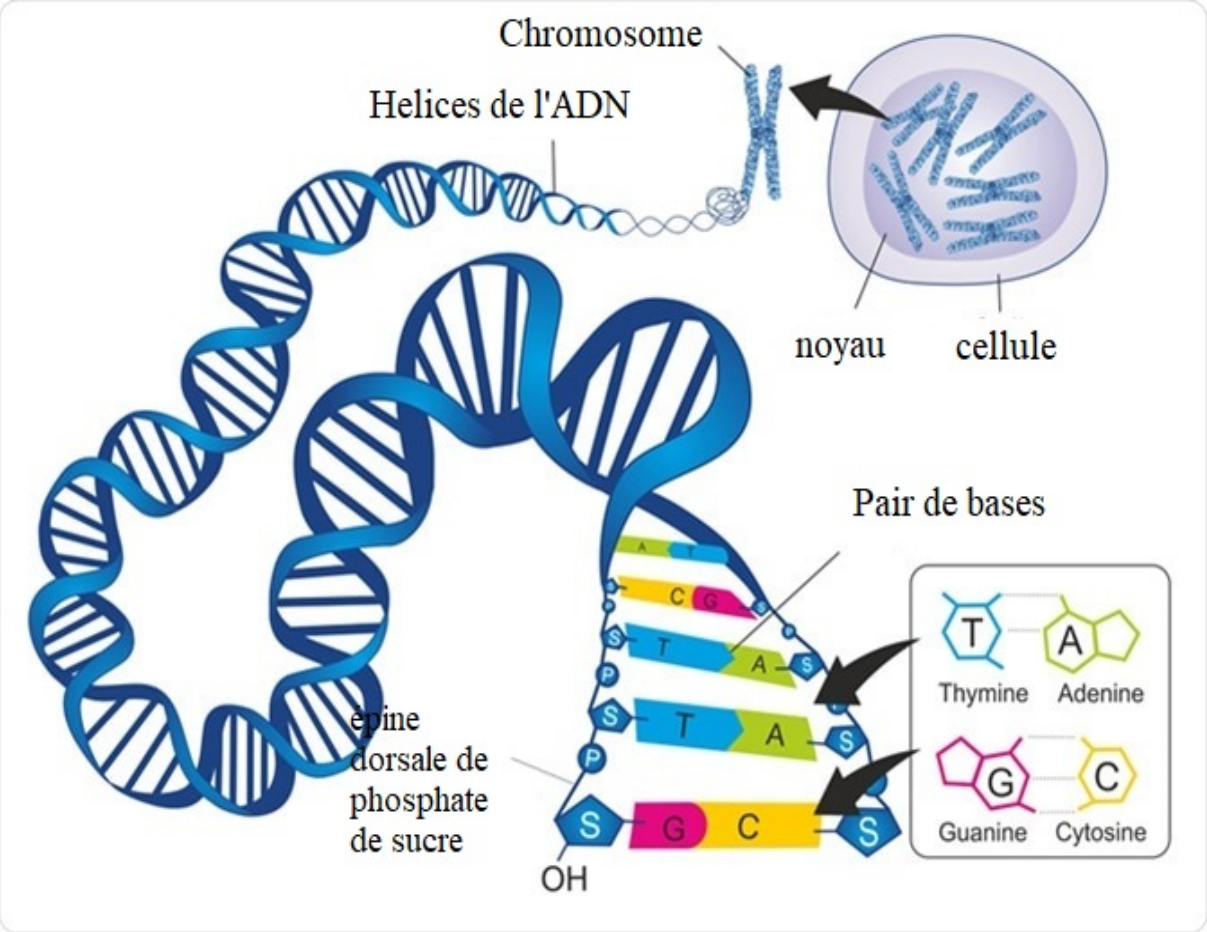


FIG. 1.1 : La structure de l'ADN dans une cellule eucaryote [1].

1.2.2 ARN

La structure fondamentale de l'ARN est identique à celle de l'ADN, à deux exceptions près : le ribose, le composant sucre de l'ARN, contient un groupe hydroxyle en position 2, et le L-uracile remplace la thymine dans l'ADN. La figure 1.2 [1] illustre la différence entre l'ARN et l'ADN [3].

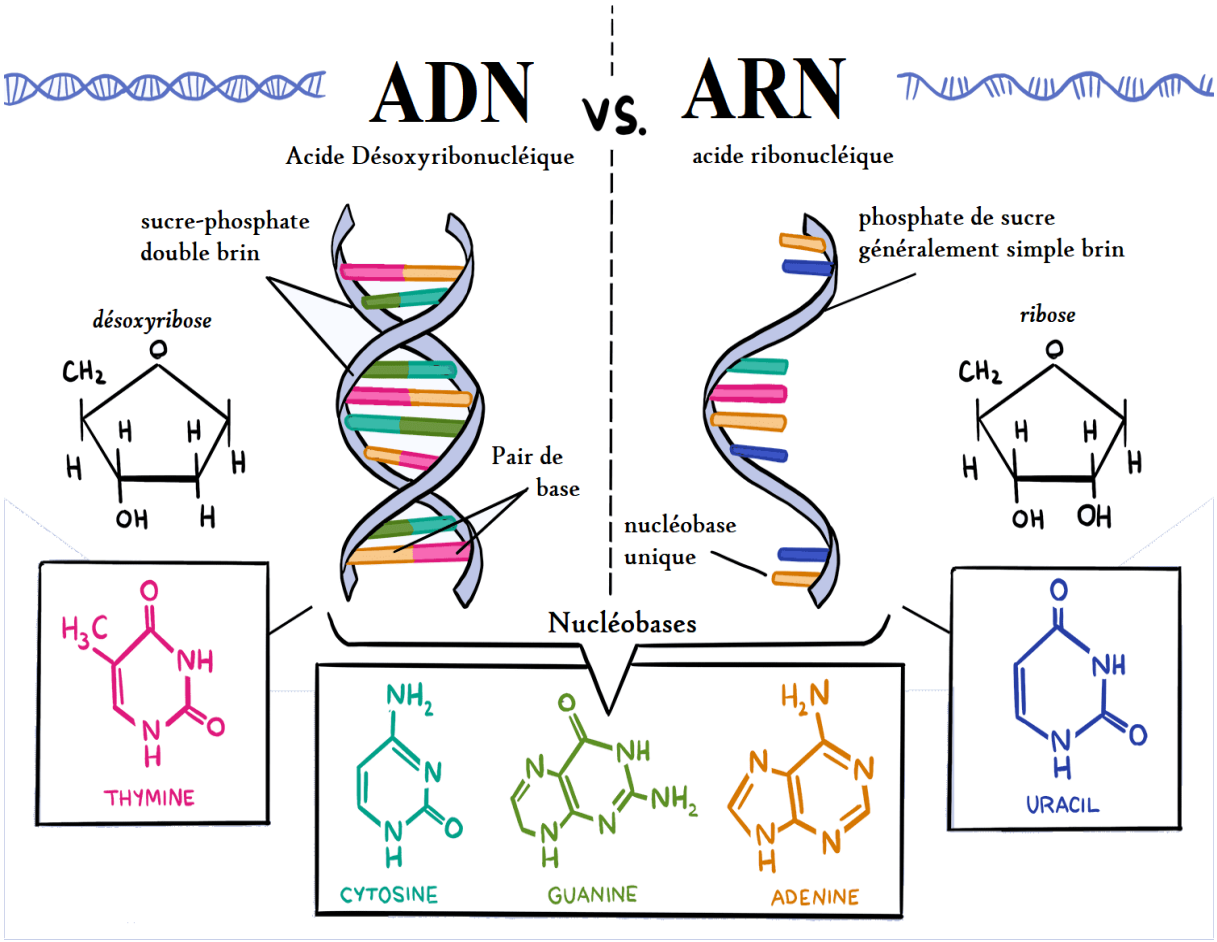


FIG. 1.2 : Les différences entre l'ARN et L'ADN dans les cellules eucaryotes.

1.2.3 Protéine

Les protéines représentent environ 60 % du contenu organique d'une cellule et sont responsables de la majorité des fonctions cellulaires, telles que les processus métaboliques, les réactions enzymatiques, le transport et les rôles régulateurs. Ils contribuent également à de nombreux composants structurels de la cellule. [4].

La structure des protéines a quatre niveaux d'organisation (voir figure 1.3 :

- **Structure primaire** .
- **Structure secondaire** .
- **La structure tertiaire** .
- **La structure quaternaire** .

Les protéines peuvent être classées en quatre catégories en fonction de leurs fonctions :

- **Les protéines de structure :** Les protéines font partie de la structure des cellules.

- **Les enzymes :** Les protéines qui catalyse les réactions.
- **Les protéines de régulation :** des protéines qui régulent l'expression d'un gène ou l'activité d'une autre protéine.
- **Les protéines de transport :** des protéines qui transportent d'autres molécules à travers les membranes ou dans le corps.

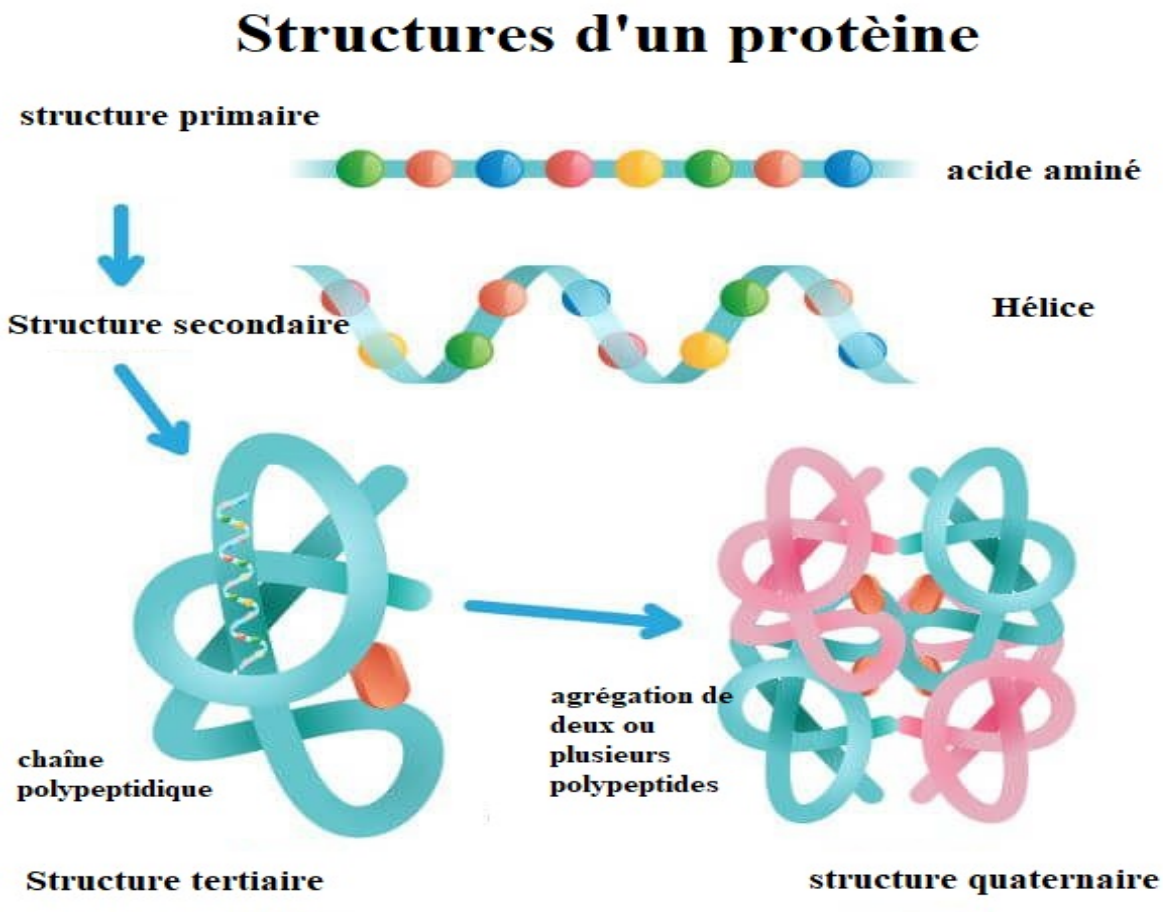


Fig. 1.3 : Les différentes structures de protéines.

1.2.4 Gène

L'unité physique et fonctionnelle de base de l'hérédité est le gène. Les gènes sont des séquences d'ADN. Certains gènes servent de modèles pour la production de molécules ressemblant à des protéines.
Cependant, alors que la plupart des gènes sont identiques chez les individus d'une même espèce, un petit nombre de gènes peuvent changer (moins de 1% du total). Différentes variantes du même gène sont appelées allèles. Ces différences mineures contribuent aux caractéristiques physiques distinctes de chaque personne. [5].

1.2.5 Génome

Chaque créature vivante a un « génome » qui contient toute son information génétique, qu'il s'agisse d'une seule cellule ou d'un virus. Hans Winkler, un botaniste allemand, a inventé le terme "génome" en 1920, combinant les expressions "gène" et "chromosome", ce qui implique que le terme génome désigne à la fois l'ensemble du génome des bases (chromosomes) et des unités génétiques (gènes). L'élément « chromosome » du mot s'est progressivement estompé et la définition du génome est devenue « toute l'information génétique d'un organisme est codée dans son ADN (ou, pour certains virus, son ARN) ». Les gènes et les séquences d'ADN non codantes sont inclus. [6].

En général, un génome est l'ensemble du matériel génétique d'un organisme ou l'ensemble des molécules d'ADN d'une cellule (génome nucléaire et génome d'organites comprenant les mitochondries et les chloroplastes d'une espèce spécifique). Les chromosomes sont les unités d'organisation. Le caryotype établit l'ordre topologique des gènes le long et à travers les chromosomes, ainsi que l'interaction entre les gènes, qui est l'essence même de l'héritage du système.

1.3 Bioinformatique

La bioinformatique est devenue l'une des approches les plus essentielles pour stocker, analyser, récupérer, intégrer et simuler des données de biologie moléculaire dans les sciences de la vie. Cela est dû à l'expansion explosive des données de séquençage et aux progrès rapides des technologies de l'information. Les disciplines impliquées dans la bioinformatique sont représentées dans la figure **??**[1].
La bioinformatique est l'application de l'informatique à la biologie et à la génétique. En d'autres termes, il s'agit d'une étude assistée par ordinateur des données biologiques au niveau génétique.

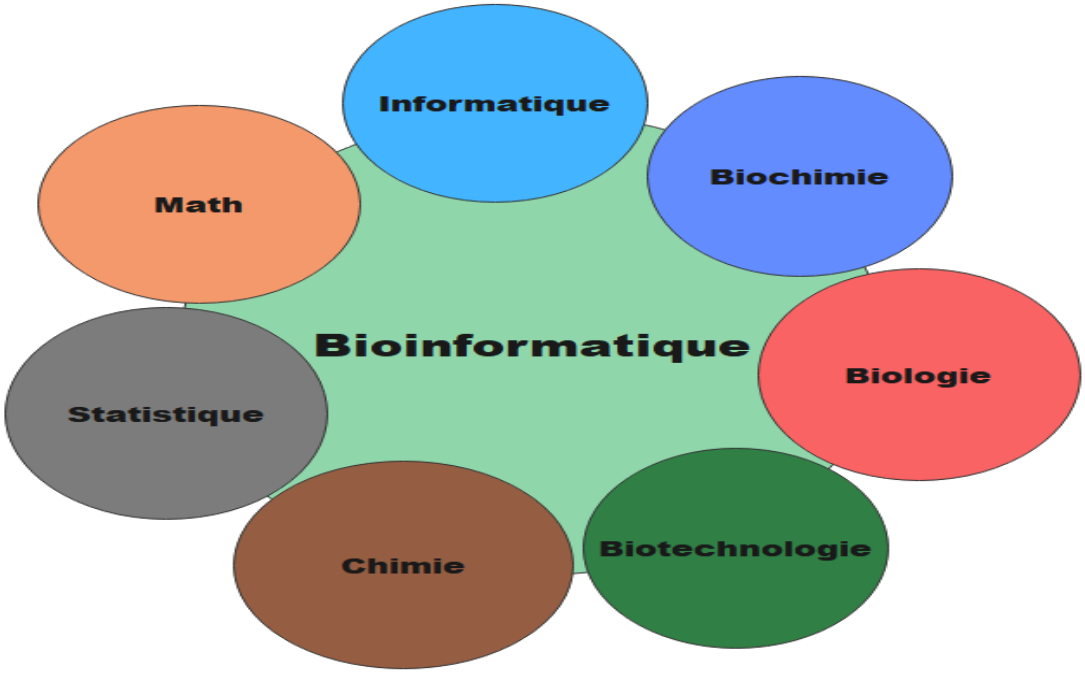


FIG. 1.4 : Disciplines participant à la bioinformatique.

Les projets de séquençage ont généré de grandes quantités de données biologiques [7] [8]. La bioinformatique nous permet de surmonter les défis du stockage, du traitement et de la représentation de ces données.

Les trois cibles importantes de la bioinformatique sont [1] :

- Traiter les données Omiques massives,
- Développer de nouvelles méthodes computationnelles compatibles avec la complexité immense de données,
- Interpréter les résultats des essais en laboratoire *invivo* et *insilico*.

Ces objectifs vont être utilisés pour résoudre ces problèmes parmi d'autres [9] :

- cartographier de différentes données (exemples : variations génétiques),
- Prédire les structures 3D des produits géniques/protéines,
- Prédire les fonctions des produits géniques/protéines,
- Concevoir des amorces,
- Comparer des séquences d'ADN/d'ARN/de protéines.

En biologie et en bioinformatique, la comparaison de séquences est un sujet crucial. Lorsque l'on tente de prédire les qualités structurelles et fonctionnelles, il s'agit d'une

première étape essentielle dans l'étude des séquences découvertes. En raison de l'avènement des technologies de séquençage à haut débit, qui créent une grande quantité de données Omics, les comparaisons de séquences sont devenues de plus en plus pertinentes. En comparant les séquences aux bases de données, cela permet la découverte de nouveaux gènes/protéines, ainsi que des inférences fonctionnelles et structurelles.

1.4 Alignement des séquences

En bioinformatique, l'alignement de séquences est un sujet d'étude qui se concentre sur la création d'outils qui comparent et détectent les points communs entre les séquences d'acides aminés, d'ADN et d'ARN à l'aide d'approches mathématiques. La similarité de séquence est utilisée pour analyser l'homologie et classer les séquences biologiques selon leur fonction et/ou leur structure, découvrir des mutations et construire des arbres métagénomiques, entre autres.

De nombreux algorithmes d'alignement sont conçus autour de deux concepts : l'alignement global et l'alignement local. Dans un alignement global, tous les caractères de la séquence sont conservés, donc toutes les longueurs sont comparées, en revanche, dans un alignement local, seules les parties les plus similaires sont alignées. Chaque concept a ses propres avantages et inconvénients[1].

1.4.1 Alignement par paire

L'alignement par paires est une technique d'alignement de deux séquences qui est utilisée pour détecter les points communs entre elles. Le but ultime de l'alignement de séquences par paires est de trouver la meilleure correspondance entre deux séquences pour maximiser la correspondance des résidus. L'alignement global et l'alignement local sont les deux types de stratégies d'alignement. [1].

1.4.2 Alignement global

L'alignement global tente à aligner les deux séquences sur toutes leurs longueurs [**polyanovsky2011comparative**].

1.4.3 Alignement local

Contrairement aux alignements globaux, les alignements locaux cherchent à localiser les chevauchements[10].

1.4.4 Programmation dynamique

L'idée de programmation dynamique employée dans le cas de deux alignements de séquences est de générer un alignement entre deux caractères en se référant au meilleur alignement des deux caractères précédents. [11]. L'alignement global proposé par Needleman–Wunsch [12] et la version locale par Smith-Waterman[13] sont les principales implémentations de cette technique.

1.4.5 Alignement multiple de séquences

L'alignement de séquences multiples (MSA) est une technique qui organise des séquences d'ADN, d'ARN ou de protéines dans un tableau dans le but d'écrire des résidus homologues de différentes séquences dans la même colonne. La séquence ancestrale contient des résidus dérivés[14].
Les procédés d'alignement de séquences multiples (MSA) comprennent une variété de techniques algorithmiques. Ces techniques peuvent être utilisées pour analyser des séquences d'ADN, d'ARN ou de protéines.

1.5 Conclusion

Pour préparer le terrain pour notre étude, nous expliquons les sujets essentiels de la biologie moléculaire, de la bioinformatique et de l'alignement des séquences dans ce chapitre.

Les approches et techniques proposées pour résoudre le problème de l'alignement de séquences multiples seront abordées dans le chapitre suivant, ainsi que les idées clés de l'intelligence artificielle.

Chapitre 2

Méthodes computationnelles et MSA

2.1 Introduction

Avec l'avènement des nouvelles technologies de séquençage, appréhender un phénomène nécessite désormais des quantités massives de données, soulignant l'importance de la bioinformatique, qui combine biologie et science des données. L'utilisation des technologies d'apprentissage automatique et d'intelligence artificielle est devenue un objectif légitime et utile. L'objectif fondamental de la bioinformatique est de comprendre les systèmes et les processus biologiques qui sont trop compliqués pour être recherchés manuellement à l'aide d'approches informatiques et de la science des données. Il relie la science des données et l'intelligence informatique aux processus biologiques. Prenons, par exemple, l'étude des séquences biologiques (ADN, ARN...).

Nous présentons dans ce chapitre les concepts de l'intelligence computationnelle.

2.2 Intelligence computationnelle

L'intelligence computationnelle est une branche d'étude concernée par la résolution de problèmes logiques et algorithmiques, ainsi que par le développement de systèmes capables d'imiter ou de remplacer les personnes. [15, 16].

"L'intelligence computationnelle" (CI) décrit la capacité d'un ordinateur à apprendre une tâche via des données expérimentales ou des observations. Il existe trois types de modèles. (voir figure 2.1) : les systèmes flous, les algorithmes d'optimisation et de recherche et l'apprentissage automatique [17].

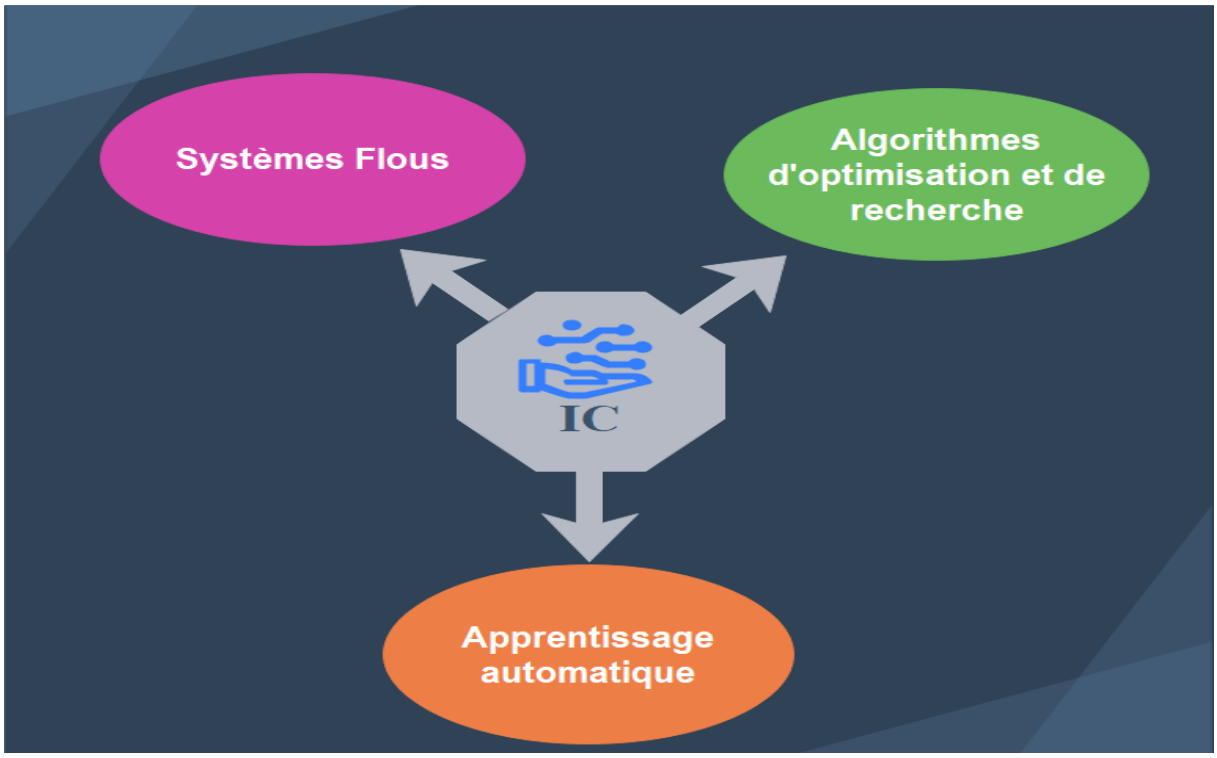


FIG. 2.1 : Les trois grandes catégories de l'intelligence computationnelle.

2.2.1 Systèmes flous

La logique floue a été utilisée avec succès dans de nombreux domaines tels que l'ingénierie des systèmes de contrôle, le traitement d'images, l'électrotechnique, l'automatisation industrielle, la robotique, l'électronique grand public et l'optimisation [1].

2.2.2 Apprentissage automatique

L'apprentissage automatique est un sous-ensemble de l'intelligence artificielle qui fait partie de l'informatique. Arthur Samuel, un employé américain d'IBM et un pionnier des jeux vidéo et de l'intelligence artificielle, a développé l'expression machine learning en 1959. [18].

L'apprentissage automatique peut être divisé en quatre groupes de base, qui sont [19, 20] : l'apprentissage automatique supervisé, l'apprentissage automatique non supervisé, l'apprentissage automatique semi-supervisé et l'apprentissage par renforcement. La figure 2.2 représentent des groupes de différents modèles d'apprentissage automatique.

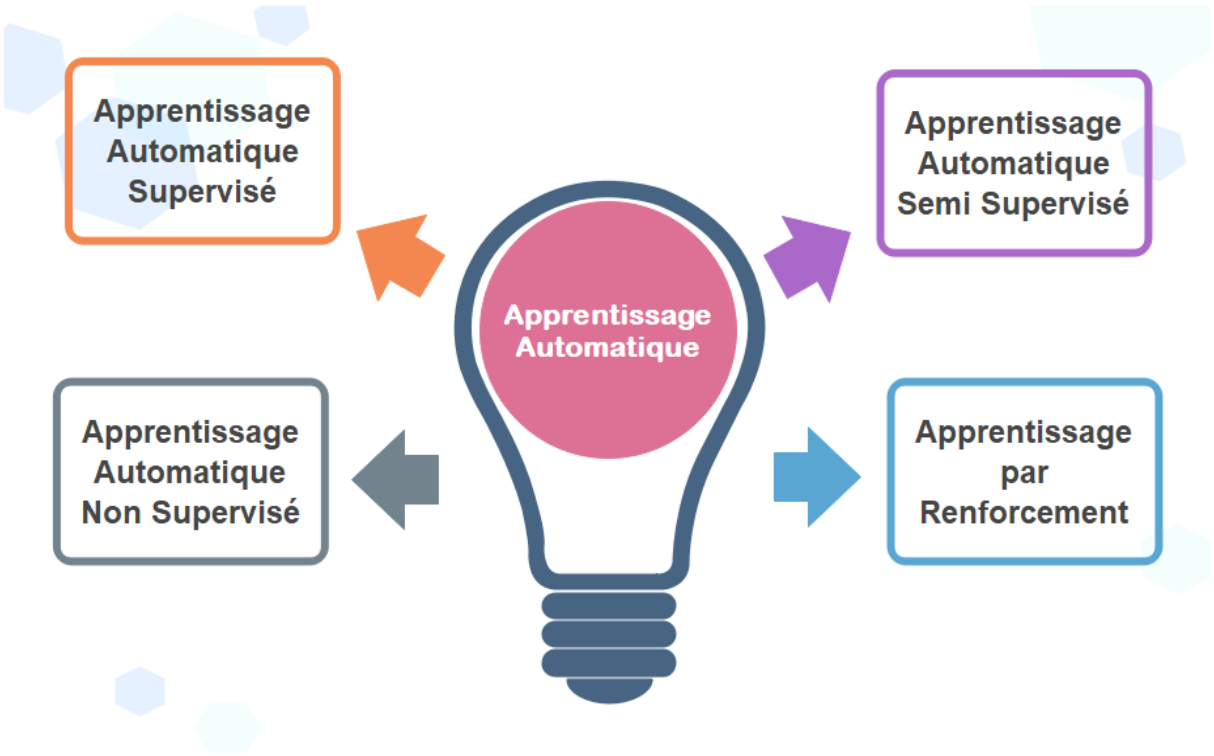


FIG. 2.2 : Les quatres catégories de l'apprentissage automatique.

Apprentissage automatique supervisé

Les approches d'apprentissage automatique supervisées utilisent des données historiques pour anticiper les évènements futurs. La partie d'apprentissage des données de cette méthode est suivie de la création d'une fonction d'inférence qui prédit la valeur de sortie. Le système peut créer des résultats basés sur les données fournies avec une formation adéquate. Pour trouver des défauts, le système compare les résultats générés aux résultats réels et ajuste le modèle en conséquence. En conséquence, les approches d'apprentissage supervisé forment un modèle pour faire des prédictions appropriées sur de nouvelles données en utilisant un ensemble de données d'entrée connues avec des résultats connus.
Le but de l'apprentissage automatique supervisé en présence d'incertitude est de construire un modèle qui permet des prédictions fondées sur des preuves. [19, 21].

Apprentissage par renforcement

L'apprentissage par renforcement profond est une catégorie d'apprentissage automatique et d'intelligence artificielle dans laquelle les robots intelligents, comme les humains, peuvent apprendre par leurs actions. Ce type d'apprentissage automatique repose sur la question de savoir si les agents sont récompensés ou pénalisés pour leurs activités. Les actions qui produisent le résultat souhaité sont récompensées. [1].

Apprentissage automatique non supervisé

L'apprentissage non supervisé est une sorte d'apprentissage automatique qui recherche des modèles auparavant inaperçus dans des ensembles de données sans étiquettes préexistantes et avec peu de supervision humaine. L'objectif est de forcer la machine à générer sa propre image à partir de l'environnement avant de développer une solution. L'auto-organisation est une propriété de ce type, par opposition à l'apprentissage supervisé, dans lequel les données sont étiquetées par des humains[22].
Il existe quatre approches de l'apprentissage automatique non supervisé [19, 22] :

- **Clustering** Le regroupement est crucial car il définit la façon dont le matériel non étiqueté est naturellement regroupé. Le regroupement peut être réalisé en utilisant une variété de méthodes, y compris : Clustering hiérarchique [23], K-moyennes [24], DBSCAN[25], etc.
- **Réseaux de neurones** incluant [26] : Auto encodeurs, Réseaux de croyance profonds…
- **Apprentissage de modèles à variables latentes** [27] incluant l'algorithme de maximisation de l'espérance, Techniques de séparation du signal aveugle…
- **Séparation des signaux aveugles** [28] basé sur l'analyse en composantes de base, l'analyse en composantes indépendantes, la factorisation matricielle non négative, la décomposition en valeur singulière[1].

.

2.3 Méthodes pour résoudre le MSA

Parce que le développement d'un MSA optimal prend du temps et de l'espace, diverses méthodes ont été conçues pour produire des solutions proches de l'idéal. Ces approches MSA sont classées en trois catégories : Méthodes exhaustives, progressives et itératives [29].

2.3.1 Méthodes exhaustives

Pour aligner des séquences simultanément, des algorithmes exhaustifs ou précis ont été imaginés. Ils peuvent construire un alignement optimal, mais seul un nombre limité de séquences peut être pris en charge. En conséquence, les principaux défauts de ces approches sont leurs besoins en mémoire importants, leur effort de calcul élevé et leur nombre limité de séquences. [30].

Généralement, les méthodes exactes sont une généralisation de l'algorithme de programmation dynamique Needleman-Wunsch [12] pour l'alignement multiple de n séquences en utilisant une matrice de score n-dimensionnelle. La valeur de chaque case

dans la matrice dépend de ses voisins, donc pour N séquences de longueur L, la taille de la matrice est $N * L$. Cette approche est tellement gourmande en termes de ressources qu'elle devient impraticable pour $N > 4$ [1].

2.3.2 Méthodes progressives

Les méthodes les plus utilisées dans la littérature sont les procédures progressives. Pour créer un MSA, les approches d'alignement progressif utilisent à plusieurs reprises des algorithmes d'alignement par paires. Ces solutions sont faciles à utiliser, rapides et ne nécessitent pas beaucoup de ressources.[29]. Étant donné n séquences de longueur L, le temps d'exécution pour l'alignement progressif est $O * (n * L^2)$. Fondamentalement, les principales étapes de ces algorithmes sont les suivantes [1] :

1. Choisir deux séquences et aligner-les ;
2. Choisir une autre et aligner-la avec le consensus des séquences précédemment alignées ;
3. Répéter l'étape 2 jusqu'à l'alignement de toutes les séquences.

Le problème dans ces méthodes est de déterminer l'ordre selon lequel les séquences sont alignées. L'une des solutions les plus prometteuses est l'utilisation de l'arbre guide suggéré par Feng et Doolittle [31]. L'alignement sera créé en utilisant l'arbre guide de la commande comme point de départ. Après que les deux séquences les plus proches ont été alignées, d'autres séquences sont ajoutées en les alignant avec l'alignement existant. Il y a aussi l'option d'alignement partiel. Même si l'arbre est excellent, l'alignement ne se traduit pas toujours par un alignement optimal. Les erreurs créées à un niveau précoce, par exemple, peuvent s'étendre aux étapes suivantes et sont impossibles à corriger. L'algorithme de Feng et Doolittle est utilisé pour développer une variété d'applications MSA, telles que CLUSTAL.[32]. La construction de l'arbre de similarité est la principale différence entre ces programmes ; par exemple, CLUSTAL utilise l'algorithme Voisin-Jointure [33] pour générer l'arbre guide. Généralement, les étapes de base de l'algorithme Feng-Doolittle sont les suivantes :

1. Calculer tous les scores possibles d'alignement par paires entre toutes les séquences ;
2. Convertir les scores des alignements en distances et construire l'arbre guide en utilisant la matrice de distance et une méthode de construction arborescente ;
3. Un alignement multiple est créé graduellement en commençant par les séquences les plus proches. Les séquences sont ajoutées une par une selon l'ordre donné par l'arbre guide.

Selon la technique d'alignement par paires utilisée, les aligneurs progressifs peuvent être globaux ou locaux. Une technique globale par paires, telle que l'algorithme Needleman, est utilisée pour aligner chaque séquence non alignée sur des séquences précédemment alignées avec des étapes précédentes dans un alignement progressif global comme

CLUSTAL. L'alignement multiple local, d'autre part, aligne uniquement les motifs les plus conservés en utilisant une approche d'alignement local telle que l'algorithme de Smith-Waterman. En général, les meilleurs résultats sont obtenus en alignant une famille de séquences avec une séquence orpheline, des séquences équidistantes et des familles de séquences divergentes à l'aide d'algorithmes globaux. En cas d'extensions importantes et d'insertions internes, cependant, les méthodes basées sur l'alignement global sont moins précises que les méthodes basées sur des algorithmes locaux [1].

2.3.3 Méthodes itératives

Les approches d'alignement itératif fonctionnent en créant un alignement initial, puis en l'affinant au cours d'une série d'itérations jusqu'à ce qu'aucune autre amélioration ne puisse être apportée. Les approches d'alignement itératif sont fondées sur la prémisse qu'une solution peut être dérivée d'une solution sous-optimale qui existe déjà.
Par exemple, PRRP [34] et MUSCLE [35] optimisent un alignement global progressif en divisant les séquences en deux groupes à l'aide d'un algorithme d'alignement de profil à profil. Dialign [36] et Dialign-TX [37] utilisent une information locale pour orienter l'alignement global. HMMT [38], un modèle Hiden Markov, utilise le recuit simulé et la programmation dynamique pour créer un alignement sous-optimal.

2.3.4 Amélioration des performances MSA dans le Big Data

Toutes les méthodes de résolution du problème MSA prennent du temps. Des recherches récentes se sont concentrées sur les moyens d'accélérer les solveurs MSA sans sacrifier la précision. Pour résoudre ce problème, diverses méthodes ont été développées, qui peuvent être divisées en deux catégories [1]. La première est basée sur l'utilisation du parallélisme par l'approche matérielle. ClustalW-MPI, par exemple, utilise des systèmes à mémoire partagée et à mémoire distribuée [39] et Parallèle T-Coffee [40]. Church et al ont fourni une conception d'algorithmes MSA sur des superordinateurs avec des processeurs parallèles et une mémoire distribuée par Church et al. [41] présente une autre parallélisation de MAFFT applicable à des milliers de séquences. D'autre part, des unités de traitement graphique sont utilisées pour accélérer les programmes MSA, comme GPU-Blast [42], G-MSA [43], et nous pouvons citer les travaux récents de Liu et al. [44] basé sur l'alignement accéléré et l'utilisation de plusieurs GPU.

Le deuxième type de parallélisme est réalisé par une méthode logicielle qui emploie des techniques de programmation parallèle. Quatre sous-classes peuvent être proposées dans cette section [45, 1] :
La première est une approche parallèle dans le calcul de la matrice de notation. Cette approche a été utilisée dans Zafalon et coll [46] où une amélioration de 15 % du temps d'exécution a été observée [45]. Le deuxième est une approche Pipeline ; Agarwal et Rizvi [47] ont proposé une technique à deux étapes pipeline qui peut améliorer la complexité du problème. Huang et coll. [48] présentent ensuite un nouveau pipeline multi-alignement pour les données de séquençage à haut débit.

La troisième est une approche parallèle avec un algorithme dynamique [45]. Dans la présente sous-classe, les techniques sont basées sur la parallélisassions des algorithmes optimaux connus sur le terrain, par exemple, la parallélisassions de l'algorithme Needleman & Wunsch par Naveed [49], la parallélisassions de l'algorithme Smith & Waterman par Dohi et al [50], et la parallélisassions de la technique optimale pour résoudre la MSA par Helal et al sur l'architecture GPU [51].

La quatrième est une technique parallèle aux données [45]. Ce type d'étude d'optimisation est devenu de plus en plus important ces dernières années. Regrouper, répartir et aligner sont les trois étapes principales de cette technique, qui est basée sur la stratégie de division pour mieux régner. La première étape propose une approche de clustering non hiérarchique, suivie d'une répartition des clusters entre les processeurs, et enfin d'une étape d'alignement pour construire le MSA [45]. Plusieurs travaux basés sur cette technique ont récemment été proposés, le premier travail est présenté par Fahad et al [52] où une technique k-mer est utilisée pour faire le regroupement, mais une perte significative de la qualité de l'alignement a été observée. Plusieurs techniques de regroupement ont été proposées, comme UCLUST [53], CD-HIT [54], BlastClust [55] et d'autres regroupements comme celui proposé dans [56]. Ceux-ci ont permis de créer différentes approches MSA. Xiangyuan et al [57] ont proposé une approche parallèle des données basée sur les deux systèmes de regroupement UCLUST et CD-HIT à l'étape de regroupement et MUSCLE à l'étape d'alignement.

2.4 Conclusion

Nous avons couvert les principes fondamentaux de l'intelligence artificielle dans ce chapitre, y compris l'apprentissage automatique et les algorithmes pour résoudre le MSA.

Dans les chapitres suivants, nous proposerons une contribution à l'amélioration du clustering, dans laquelle nous appliquerons l'intelligence computationnelle avec l'algorithme de Clustering K-means.

partie II

Contribution

Chapitre 3

Méthode de clustering MSA

3.1 Introduction

Le défi et les solutions dans le domaine du Big Data ont été explorés dans la section précédente. La majorité des méthodes MSA actuelles échouent dans les Big Data, les rendant obsolètes.

L'accumulation de données de séquence biologique dépasse les améliorations de l'efficacité du traitement, ce qui nécessite le développement de technologies améliorées à l'haut débit. La classification des séquences (clustering) est une première étape courante dans l'analyse computationnelle des séquences. Le regroupement est utilisé pour prédire l'homologie et la fonction, réduire la redondance et générer des sous-ensembles qui peuvent être traités pour des méthodes plus intensives en calcul comme l'alignement multiple de séquences biologiques, la comparaison de données provenant de différents environnements et la quantification de la diversité des écosystèmes. Une technique de recherche de séquence comme UClust [53], qui a été largement adoptée pour sa grande vitesse et sa sensibilité, est au cœur de la plupart de ces approches de catégorisation.

Dans ce chapitre, nous nous concentrons sur le k-means en utilisant les k-mers comme métrique et en utilisant l'alignement de séquences multiples pour l'évaluation pour le cas de traitement de séquences biologiques à grande échelle. L'objectif de cette étude est l'amélioration de l'évolutivité et du temps d'exécution dans le domaine de clustering des séquences. Nous proposons une stratégie basée sur l'algorithme k-means en utilisant les k-mers afin de regrouper les séquences biologiques.

Des expériences sur un ensemble diversifié de données ont démontré l'efficacité de la méthode proposée et sa capacité à atteindre un bon équilibre temps/qualité. Les tests démontrent également que notre technique peut être utilisée dans des systèmes parallèles pour accélérer l'analyse de grandes quantités de données biologiques.

3.2 Méthode proposée

Dans cette partie, nous proposons une nouvelle technique de clustering des séquences biologiques basée sur l'algorithme k-means en utilisant les k-mers afin d'améliorer le temps d'exécution dans le cas de traitement de données à grande échelle tel que l'alignement multiple des séquences biologiques.

Notre approche repose sur l'extraction des k-mers à partir des séquences biologiques, puis les utiliser dans l'algorithme k-means afin de classifier les différentes séquences dans de différents groupes.

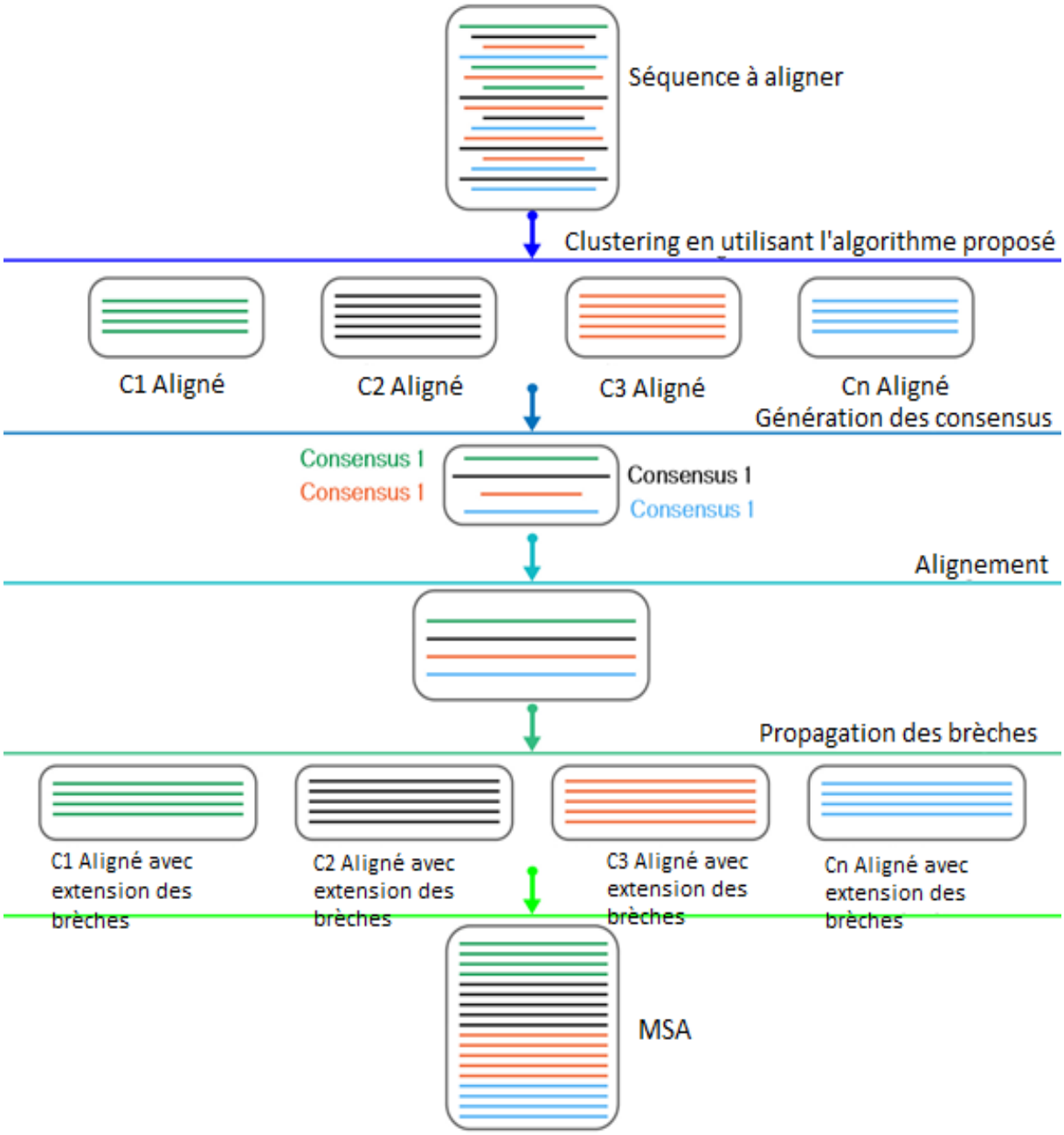


FIG. 3.1 : Aperçu de l'approche utilisé HClustMSA [45]

3.2.1 Clustering des séquences

Dans cette étape, nous allons regrouper les séquences biologiques en utilisant k-means. L'ensemble de séquences E_{seq} est définit comme suit :

$$\mathrm{E}_{seq} = \{seq_1, seq_2, ..., seq_n\}$$

Le regroupement des séquences facilite l'alignement, la prédiction de l'homologie et elle réduit la redondance, en générant des sous-ensembles $SubD^{i}_{\{}seq\}$ définis comme suit :

$$\mathrm{SubD}^{1}_{seq}, SubD^{2}_{seq}, ..., SubD^{n}_{seq}$$

Cette étape réduit le coût de calcul en comparant des séquences provenant de diverses sources et en quantifiant leur variété. Nous allons exécuter le regroupement de séquences dans notre méthode proposée.

Les étapes appliquées afin de générer des sous-ensembles sont comme suit :

1. Lire l'ensemble de séquences E_{seq} ;
2. extraire les k-mers en utilisant l'algorithme Fasta2k-mer [58] ;
3. Définir le nombre de cluster ¨N_c en se référant au nombre de cœur de CPU (minimum $N_c = 2$;
4. utiliser l'algorithme k-means pour créer les différents clusters.

Afin de mesurer la qualité du clustering, on a fait une comparaison entre notre méthode et UClust en les intégrant dans un processus d'alignement de séquence.

3.2.2 Alignement de clusters

Cette étape introduit une stratégie d'alignement progressif basée sur deux algorithmes : l'alignement global basé sur Needleman et Wunsch et l'alignement local basé sur Smith et Waterman [59]. Dans notre approche, l'alignement de chaque cluster $SubD^{n}_{seq}$ est traité comme suit :

1. Initialiser k une différence limite ;
2. Assigner le cluster $SubD^{n}_{seq}$ à un cœur CPU ;
3. Construire une arbre phylogénétique $Tree_{SubD^{n}_{seq}}$ en utilisant la méthode ClustalW [32] ;
4. Choisir les deux séquences Seq_i et Seq_j les plus proches dans l'arbre $Tree_{SubD^{n}_{seq}}$;
5. Calculer la différence $diff(seq_i, seq_j)$ entre les deux séquences Seq_i et Seq_j en utilisant l'équation 3.1 ;

$$\mathrm{diff}(\mathrm{Seq}_i, Seq_j) = Taille(Seq_i) \div Taille(Seq_j) \quad (3.1)$$

6. Si $diff(seq_i, seq_j) <$ k ; utiliser l'alignement global de Needleman et Wunsh ;
7. Sinon ; si $diff(seq_i, seq_j) \geq k$; utiliser l'alignement local de Smith et Waterman ;
8. Propager les gaps ;
9. Répéter *(4)*.

Nous obtiendrons un MSA pour chaque cluster à la fin de cette étape, qui construira une séquence de consensus pour la phase suivante.

Les étapes de l'alignement des clusters sont décrites dans le pseudo code 1

Algorithm 1 Pseudo code de la phase de l'alignement des séquences

Require : $SubD^1_{seq}, SubD^2_{seq}, ..., SubD^n_{seq}, k$
Ensure : $MSA_{SubD^1_{seq}}, MSA_{SubD^2_{seq}}, ..., MSA_{SubD^n_{seq}}$
for all $SubD^n_{seq}$ **do**
Tree $\leftarrow ArbrePhylogenetique(SubD^n_{seq})$ **while** $\neg ConditionArret()$ **do**
$seq_a, seq_b \leftarrow PlusProche(Tree)$
$diff \leftarrow CalculerDifference(seq_a, seq_b)$
if $diff < k$ **then**
$AlignementGlobal(seq_a, seq_b)$
else
$AlignementLocal(seq_a, seq_b)$
end if
end while=0

3.3 Résultats expérimentaux

L'approche proposée est implémentée dans MATLAB R2014b et nous avons effectué ces tests sur un Intel Core I5 8250U Avec 4 cœurs, fonctionnant à 1,60 GHz, avec des caches (L1D-Cache 32 KB, L1I-Cache 32 KB, L2-Cache 256 KB, L3-Cache 6 MB) et avec 8 Go de mémoire DDR4. Plusieurs expériences ont été réalisées pour étudier le gain fourni par la stratégie proposée.

3.3.1 Temps d'exécution

Nous avons créé un jeu de tests basé sur de grands ensembles de données contenant des jeux de référence de séquence générés par GenRGenS sur des profils de séquence réels dérivés de BAliBASE dans lesquels la longueur d'une séquence varie de 500 à 2000 et le nombre de clusters est fixé au nombre de processeur pour mesurer le gain fourni par notre approche, qui utilise une étape de regroupement avant l'alignement. Le temps d'exécution de MSA est directement proportionnel au nombre de séquences et à leur longueur.

Nous avons créé l'approche identique, avec et sans regroupement comme étape initiale avant l'alignement, pour évaluer les techniques en termes de temps d'exécution. Nous

avons également utilisé ClustalW comme technique d'alignement pour comparer notre réponse à UClust.

L'approche proposée est implémentée dans MATLAB R2014b et nous avons effectué ces tests sur un Intel Core I5 8250U Avec 4 cœurs, fonctionnant à 1,60 GHz, avec des caches (L1D-Cache 32 KB, L1I-Cache 32 KB, L2-Cache 256 KB, L3-Cache 6 MB) et avec 8 Go de mémoire DDR4.

La figure 3.2 montre la différence de temps d'exécution entre la technique identique avec et sans clustering ; le clustering peut réduire considérablement le temps d'exécution.

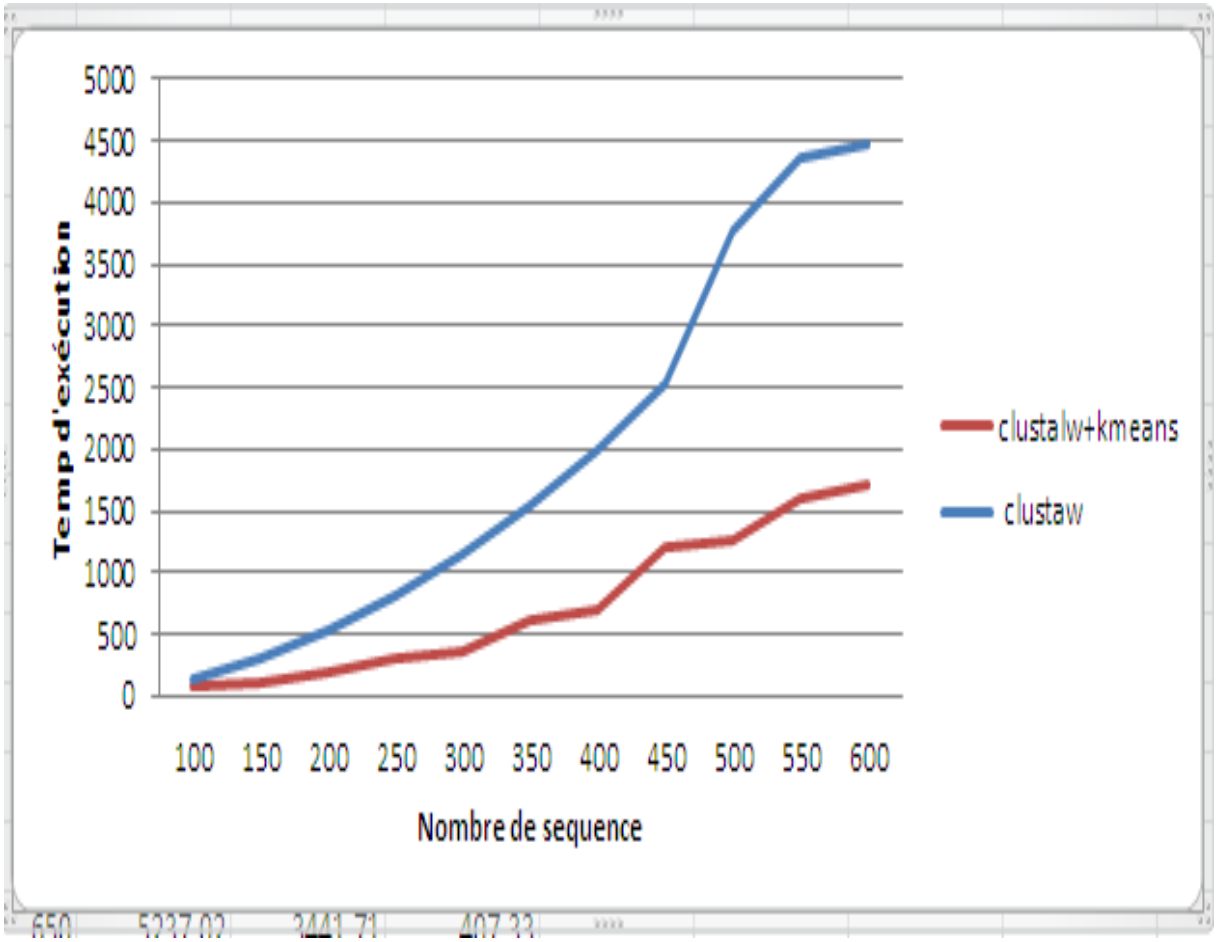
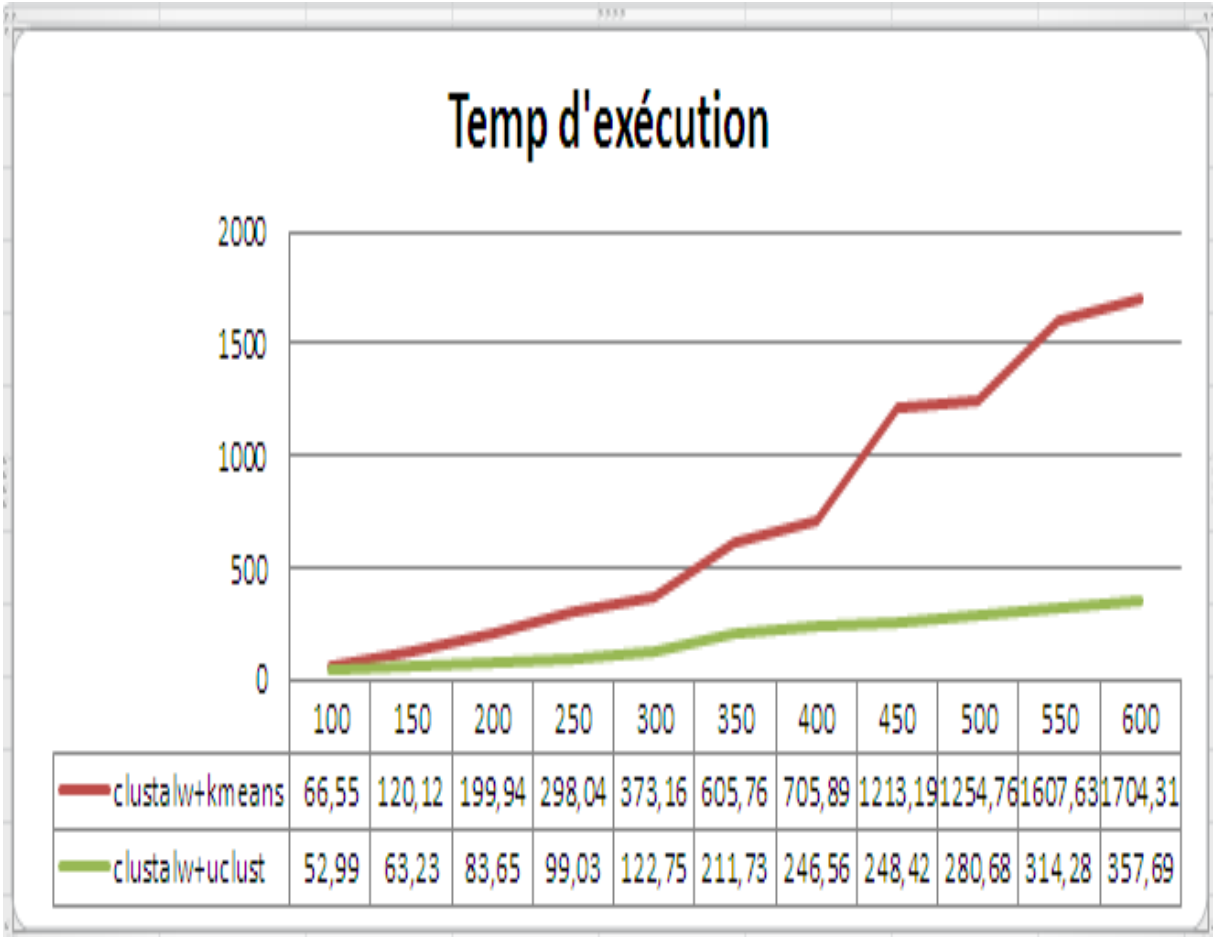


FIG. 3.2 : La différence de temps d'exécution utilisant clustalw avec et sans clustering.

La figures 3.2 et comme le clustering est utilisé, il y a une amélioration considérable du temps d'exécution par rapport à son non utilisation.
La figure 3.3 montre que UClust est très rapide par rapport à k-means, dans le cas d'ensembles de données à grande échelle.



	100	150	200	250	300	350	400	450	500	550	600
clustalw+kmeans	66,55	120,12	199,94	298,04	373,16	605,76	705,89	1213,19	1254,76	1607,63	1704,31
clustalw+uclust	52,99	63,23	83,65	99,03	122,75	211,73	246,56	248,42	280,68	314,28	357,69

FIG. 3.3 : Comparaison des temps d'exécution entre kmeans et uclust en utilisant clustalw.

3.3.2 Qualité de l'alignement

Pour étudier la qualité de l'alignement, nous avons fait une comparaison des performances de BaliBase 3.0 en utilisant ClustalW sans clustering et avec clustering (ClustalW+Kmeans et ClustalW+uclut).

Pour mesurer la qualité de l'alignement, le programme qScore [60] a été utilisé. Le score PREFAB Q (également appelé score Balibase SPS ou score Développer) et le score Balibase TC sont générée.

En termes de qualité d'alignement et selon le Tableau 3.1 , l'alignement perd considérablement de qualité avec Uclust .

3.4 Conclusion

Nous avons présenté une nouvelle technique de regroupement pour l'alignement de séquences multiples en bioinformatique dans ce chapitre. L'alignement dans notre méthode

Tab. 3.1 : Comparaison des performances de BaliBase 3.0

Méthodes	*BB11*	*BB12*	*BB20*	*BB30*	*BB40*	*BB50*
ClustalW	0.589	0.886	0.886	0.773	0.789	0.769
	0.325	0.759	0.339	0.383	0.398	0.363
ClustalW+Kmeans	0.516	0.868	0.886	0.687	0.682	0.646
	0.292	0.717	0.339	0.234	0.260	0.225
ClustalW+Uclust	0.359	0.831	0.834	0.673	0.523	0.600
	0.168	0.653	0.205	0.203	0.119	0.169

consiste en deux étapes principales : la première est le regroupement des séquences en sous-ensembles, ce qui entraine une réduction significative du temps d'exécution pour les ensembles de données de séquences à grande échelle. Cela permet également l'utilisation naturelle du parallélisme lors de l'utilisation d'ordinateurs multi cœurs, ainsi que d'éviter l'échec des approches MSA pour aligner un volume élevé de séquences.

Notre clustering est basée sur l'algorithme K-means utilisant une table des k-mers considéré comme métrique de distance entre les différentes séquences.
On a fait des comparaisons sur la base de données BaliBAse entre notre approche et l'algorithme de clustering UCLust en les intégrant dans la procédure d'alignement dans les méthodes de parallélisassions des données.
Notre technique était très bénéfique par rapport à UClust qui perte considérablement en terme de qualité d'alignement.
Malgré les améliorations considérables obtenues en termes de temps d'exécution et la préservation en qualité dans l'alignement des séquences, notre technique perd en temps par rapport à UClust ce qui nous pousse vers la proposition d'une solution parallèle.

Chapitre 4

Implémentation

4.1 Introduction

Suite à l'aperçu théorique des chapitres précédents, nous présentons l'aspect pratique de notre application. Notre but est la réalisation d'une application capable d'aligner un ensemble de séquences biologiques d'ADN ou de protéines en essayant d'obtenir un bon temps/qualité d'alignement.
Notre application est une extension de l'application ACE-MSA [59] avec notre approche de clustering proposée. L'extension consiste en l'intégration de l'implémentation de K-means avec K-mers en MATLAB.

4.2 Présentation du langage de programmation

4.2.1 Matlab

MATLAB est un système de programmation scientifique interactif pour le calcul numérique et la visualisation graphique, basé sur la représentation matricielle des données, qui tire son nom du Matrix Lab. C'est un outil multiplateforme disponible pour les environnements Windows, Unix (et dérivés BSD, Linux, Solaris, MacOs...). [61].

4.2.2 La base utilisée

Nous avons utilisé dans notre application la troisième version du benchmark d'alignement multiple, le plus largement utilisé BAliBASE, fournissant un manuel de haute qualité d'alignement de référence raffiné, basé sur la structure 3D des protéines. BAliBASE 3.0 inclut de nouveaux cas de test plus difficiles, représentant des problèmes réels rencontrés lors de l'alignement de séquences complexes. [62].

4.2.3 Format Fasta

Une séquence au format FASTA est composée de deux parties. Une représente l'entête qui distingue les séquences par le signe (">") suivie d'un identifiant et du commentaire, et une représente la séquence. Exemples de séquences au format fasta (Figure 4.1).

```
>PROA_CLOTE
MDDLNKYLINKGKKAKEASRFLSSVDSNFKNKALHKMGEDLKANMNKIIAANKIDMEKGKEKGLSKSLLDRLLIDEKRVN
DMVNGLIEVAELPDPIGEVLNMWKRPNGINIGVKRVPLGVIGIIYEARPNVTVDATALCLKSGNAVILRGGSEAINTNKA
IGKILENSAIESGLPEGTIQLIETTDREIVNKMLKLNEYIDVLIPRGGRGLIDNVVKNSTVPVIQTGVGLCHVYVDGSAN
LKMAQDIIVNAKTQRPGVCNALETLLVHKDVANSFLPEIVSEISKYGVESKLCEKSFEVVKGSIKDAKVLSLISEATEED
WDTEYLDLILSIKIVNSLDEALNHIYDHGTKHSEAIITENYTNSQRFLNEVDAAAVYVNASTRFTDGSQFGFGAEIGIST
QKLHARGPMGLTQLTTTKYIIYGNGQIR
```

Fig. 4.1 : Format Fasta

4.3 Description de l'interface

L'interface graphique est composée en deux parties (Figure4.2) : une pour faire l'alignement multiple de séquences sur un fichier fasta (Figure 4.3), et une pour calculer la qualité en comparant les séquences alignées avec des séquences référencé. La figure 4.6) montre l'interface principale de notre application.

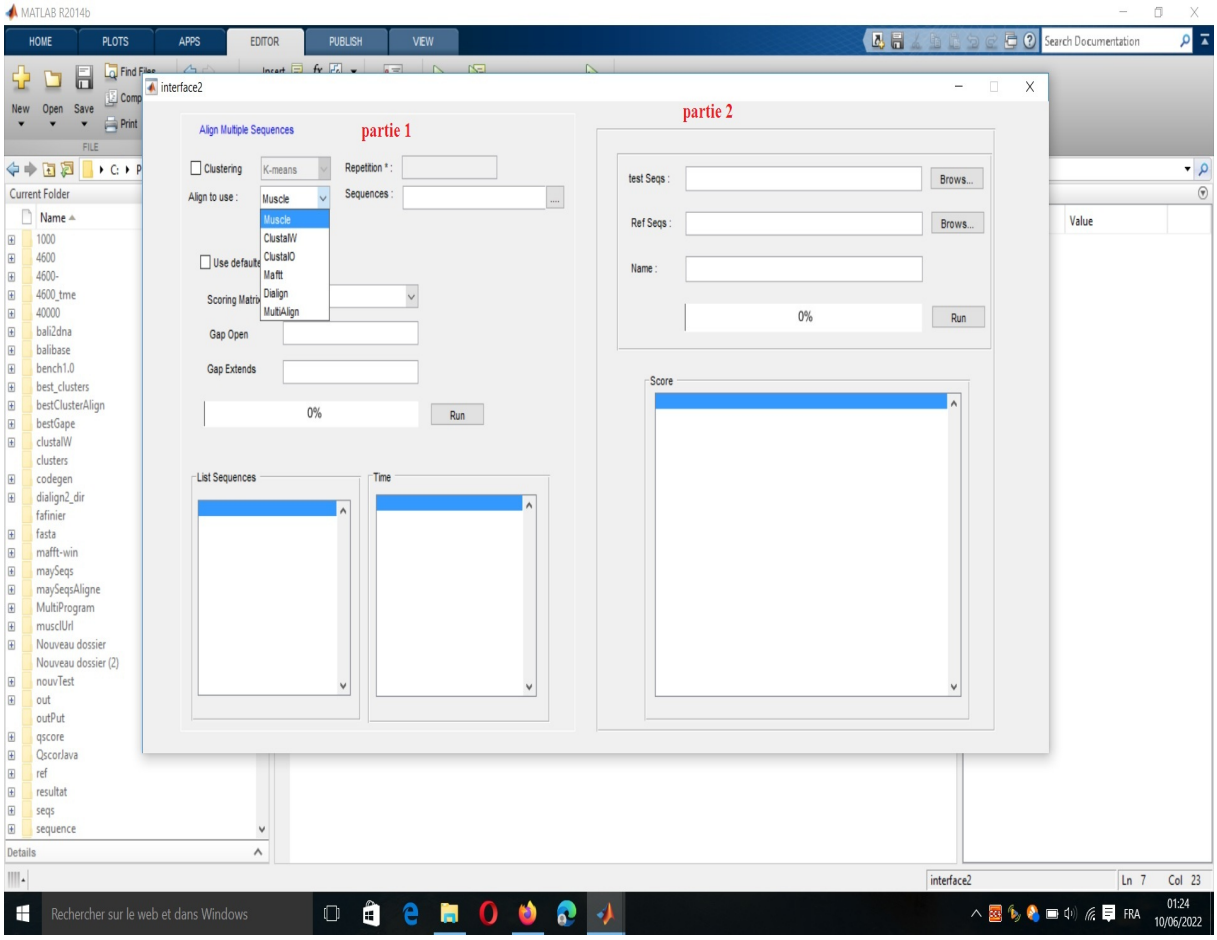


FIG. 4.2 : Interface principale

4.3.1 L'alignement multiple de séquences

Avant de lancer l'alignement il faut choisir l'algorithme d'alignement. Le clustering est optionnel, on peut choisir d'utiliser notre technique basée k-means.

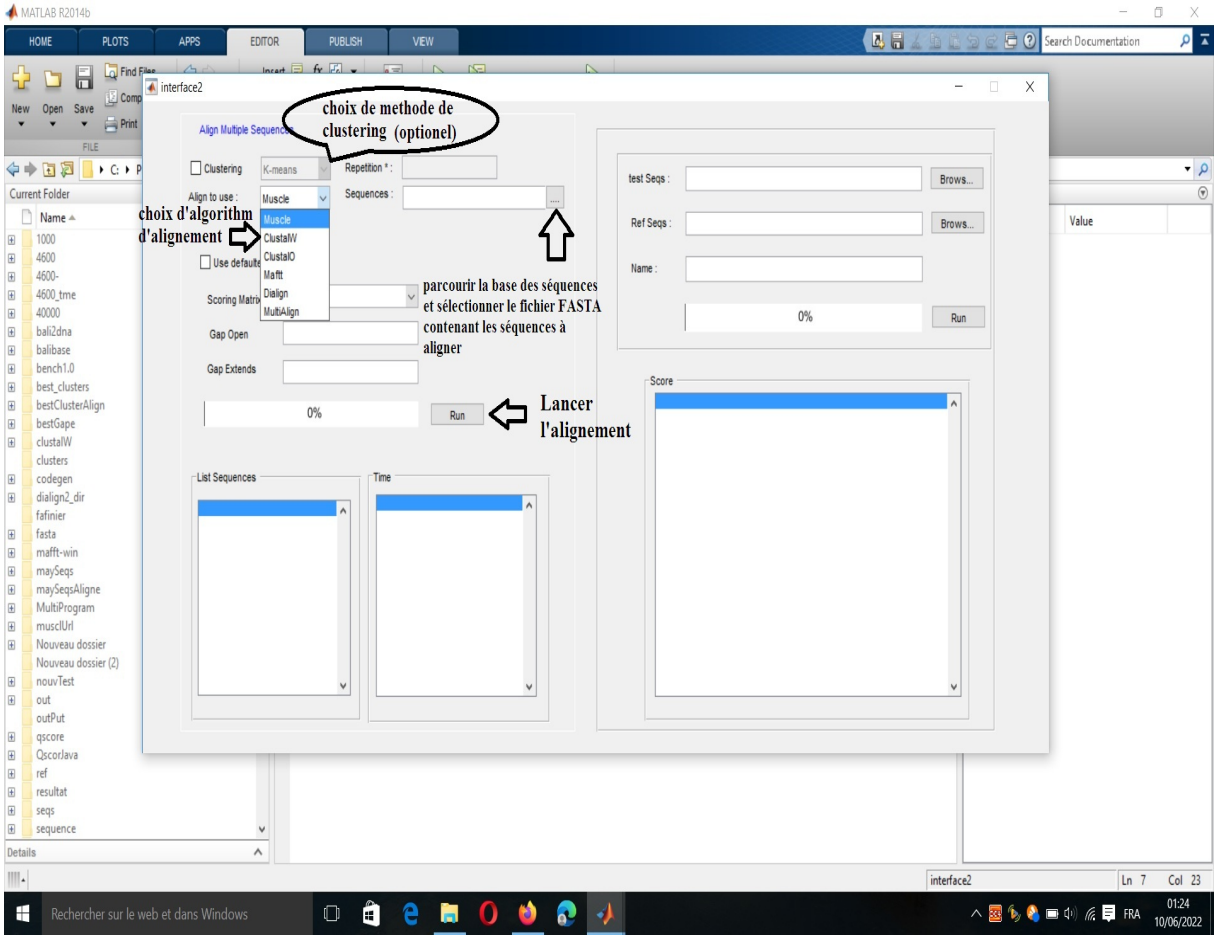


FIG. 4.3 : Partie d'alignement

Lorsque l'alignement est terminé, le résultat d'alignement multiple de chaque fichier est affiché et enregistré dans un fichier nommé : time.csv (Figure 4.4).

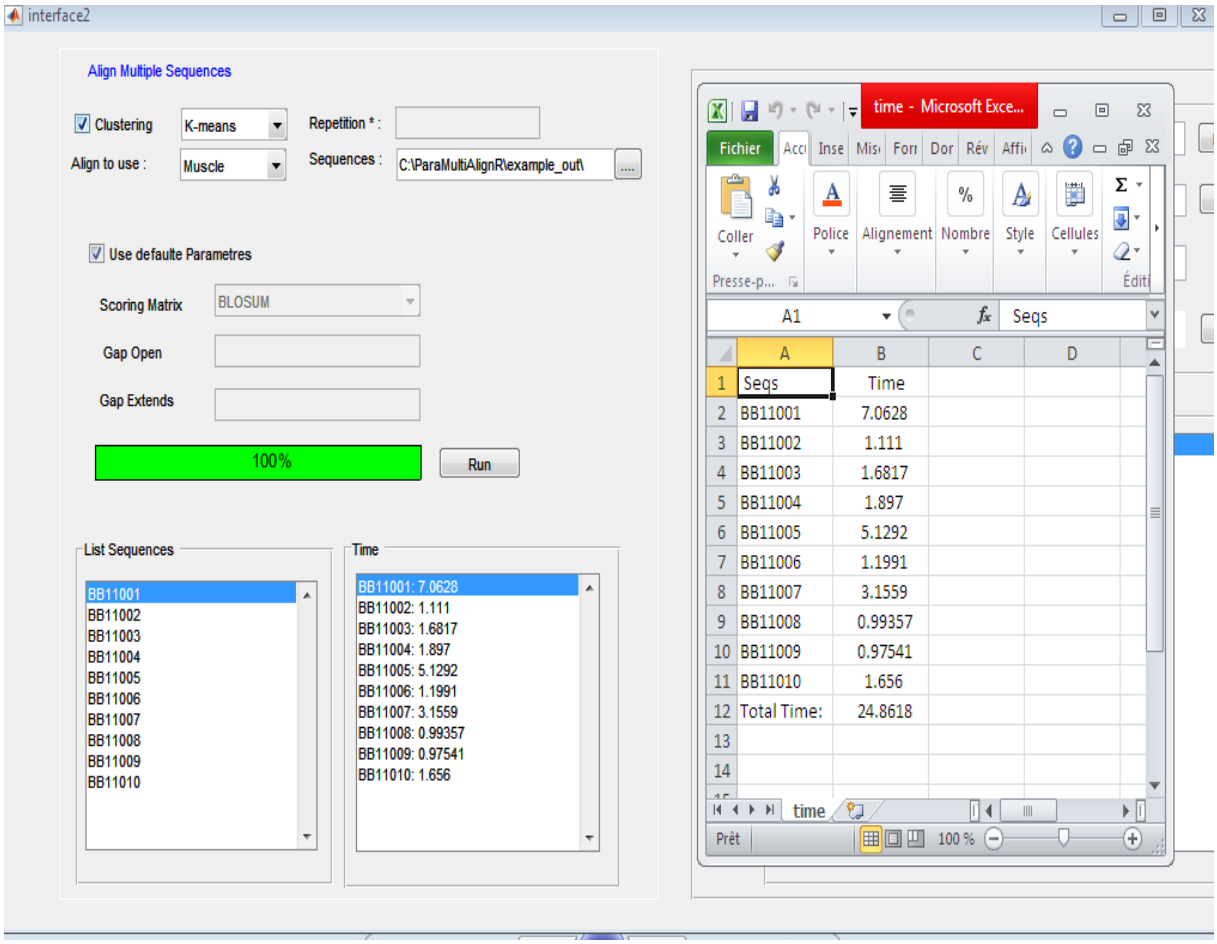


FIG. 4.4 : Fin d'alignement

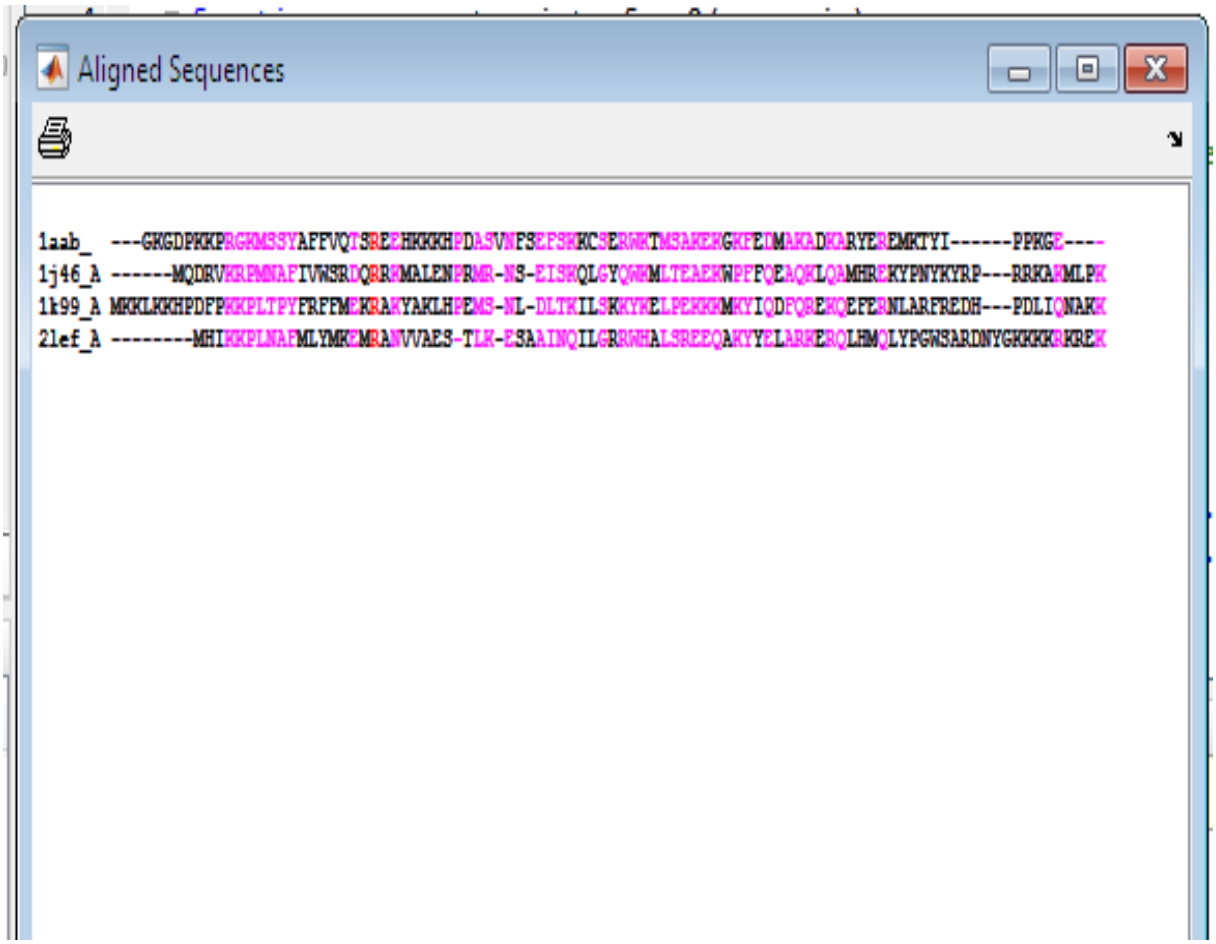


Fig. 4.5 : Résultat d'alignement d'un fichier

4.3.2 Calculer la qualité d'alignement multiple de séquences

Avant de lancer le calcul, il faut parcourir les bases des séquences à aligner, puis on choisit les séquences de référence et ainsi mesurer la qualité d'alignement en utilisant QScore. 4.6)

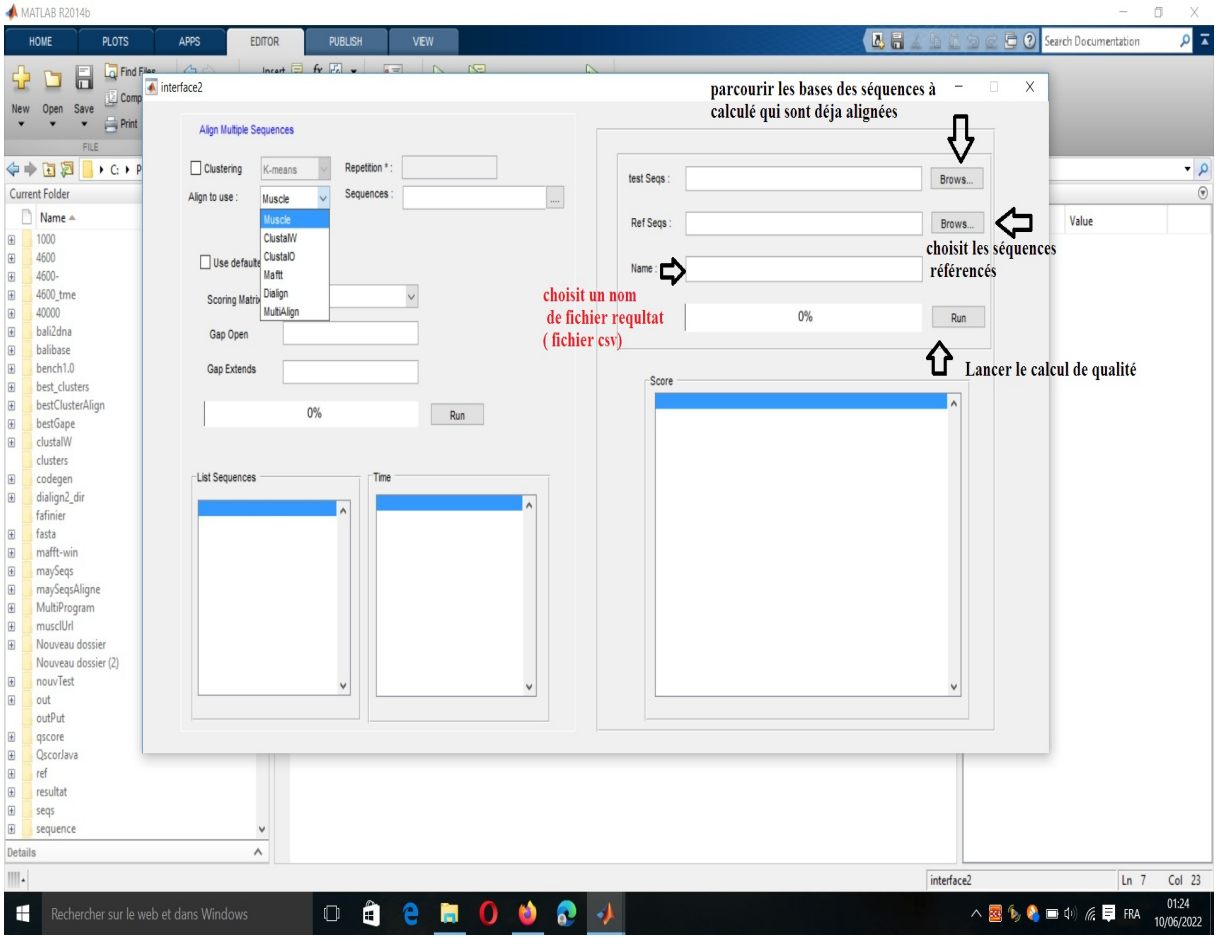


Fig. 4.6 : Partie 2 la qualité

À la fin du calcul, les résultats sont affichés et enregistrés dans un fichier csv (Figure 4.7).

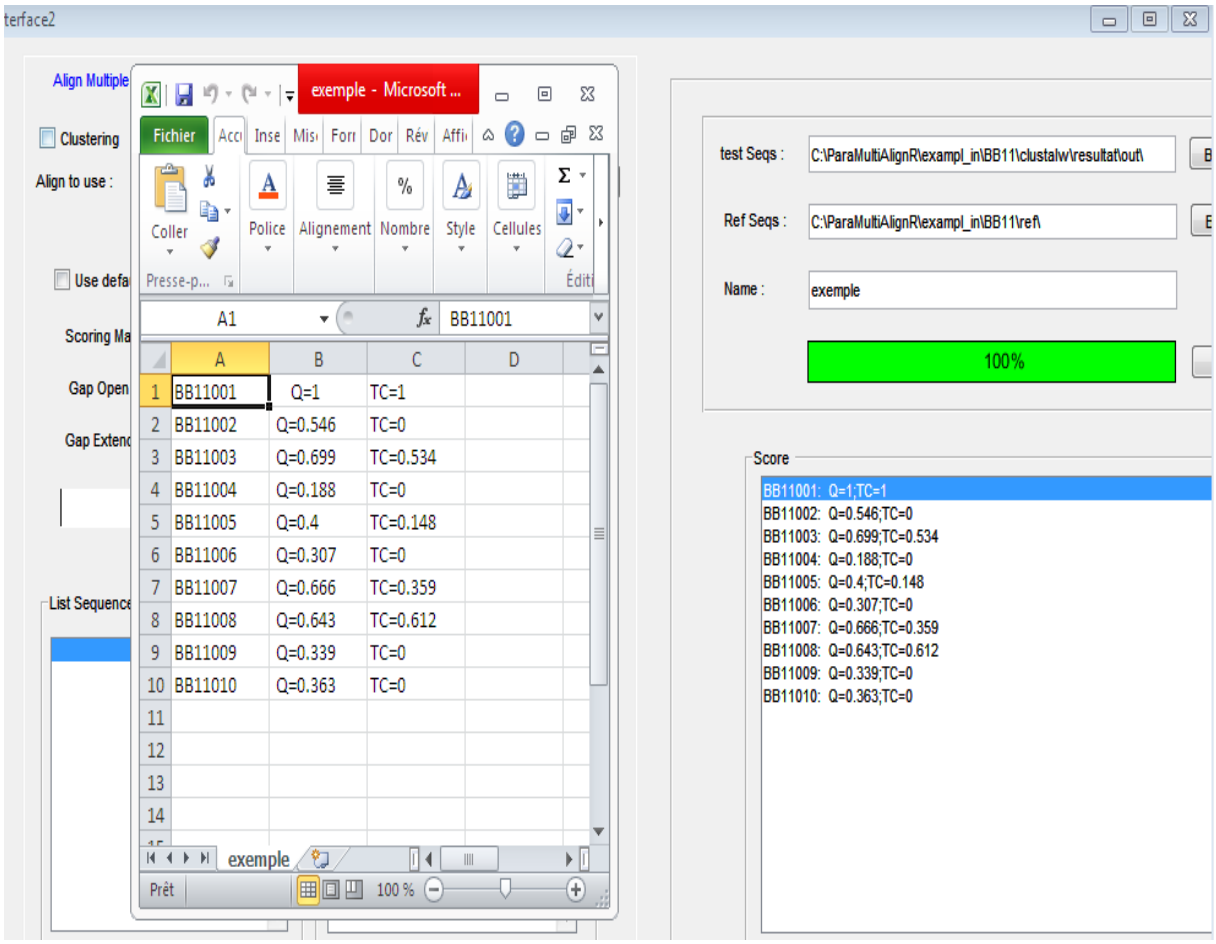


Fig. 4.7 : Résultat de calcul de la qualité

4.4 Coclusion

Dans ce chapitre nous avons présenté les fonctionnalités de notre application pour l'alignement des séquences biologique en utilisant plusieurs algorithmes d'alignement. L'utilisation de clustering joue un rôle important pour optimiser le temps d'exécution. K-means a été utilisé comme algorithme de clustering ce qui donne des résultats satisfaisables en préservant la qualité d'alignement. Notre application donne aussi des résultats de la qualité d'alignement en comparant les séquences alignées avec les séquences de référence en utilisant QScore.

Conclusion et perspectives

Conclusion générale

L'avancement rapide des technologies de séquençage à haut débit a poussé la recherche sur le traitement des données biologiques dans le domaine du Big Data. De nombreux outils bioinformatiques doivent être mis à jour afin de traiter ces quantités massives de données.

Nous avons abordé le défi de l'alignement multiple de séquences biologiques dans le contexte du Big Data dans ce mémoire. Des techniques basées sur le paradigme divisé pour mieux régner sont utilisées. Avant la phase d'alignement, toutes ces approches réalisent une étape de regroupement de séquences biologiques. L'objectif est de réduire le temps de calcul tout en préservant la qualité de l'alignement.

Dans notre travail, nous avons développé une technique de clustering basée sur l'algorithme k-means en utilisant les k-mers liées aux séquences biologiques. Nous avons testé la technique sur les différents Benchmarks de la base BAliBASE. Dans son implémentation, notre technique a montré de très bons résultats en termes de qualité tout en perdant en temps de calcul. La technique était gourmande par rapport à UClust.

Perspectives

L'utilisation de notre clustering était très bénéfique en terme de préservation de qualité et d'amélioration en temps d'exécution, toute fois, l'utilisation de UClust rend l'algorithme global plus rapide que le nôtre ce qui nous pousse à créer une version distribuer utilisant les différentes plateformes de parallèlisation comme le Spark- Appach [63].

Bibliographie

1. LEBSIR R. Doctorat - Constantine 2. 2022
2. Lyubchenko YL. DNA structure and dynamics. Cell biochemistry and biophysics 2004 ; 41 :75-98
3. Eddy SR. Computational genomics of noncoding RNA genes. Cell 2002 ; 109 :137-40
4. Somero GN. Proteins and temperature. Annual review of physiology 1995 ; 57 :43-68
5. Ruelland JG. L'Empire des gènes : histoire de la sociobiologie. ENS éditions, 2004
6. Gelderblom HR. Structure and classification of viruses. Medical Microbiology. 4th edition 1996
7. Stein L. Genome annotation : from sequence to biology. Nature reviews genetics 2001 ; 2 :493-503
8. Erickson JW et Altman GG. A search for patterns in the nucleotide sequence of the MS2 genome. Journal of Mathematical Biology 1979 ; 7 :219-30
9. Baxevanis AD, Bader GD et Wishart DS. Bioinformatics. John Wiley & Sons, 2020
10. Polyanovsky VO, Roytberg MA et Tumanyan VG. Comparative analysis of the quality of a global algorithm and a local algorithm for alignment of two sequences. Algorithms for molecular biology 2011 ; 6 :1-12
11. Gollery M. Bioinformatics : sequence and genome analysis. Clinical Chemistry 2005 ; 51 :2219-20
12. Likic V. The Needleman-Wunsch algorithm for sequence alignment. Lecture given at the 7th Melbourne Bioinformatics Course, Bi021 Molecular Science and Biotechnology Institute, University of Melbourne 2008 :1-46
13. Sandes EFdO et Melo ACM de. Smith-waterman alignment of huge sequences with gpu in linear space. *2011 IEEE International Parallel & Distributed Processing Symposium*. IEEE. 2011 :1199-211
14. Singh D et Reddy CK. A survey on platforms for big data analytics. Journal of big data 2015 ; 2 :1-20
15. Raj JS. A comprehensive survey on the computational intelligence techniques and its applications. Journal of ISMAC 2019 ; 1 :147-59
16. Chen L, Chen P et Lin Z. Artificial intelligence in education : A review. Ieee Access 2020 ; 8 :75264-78

17. Andries P et al. Computational intelligence : an introduction. 2022

18. Park C, Took CC et Seong JK. Machine learning in biomedical engineering. Biomedical Engineering Letters 2018 ; 8 :1-3

19. Hutter F, Kotthoff L et Vanschoren J. Automated machine learning : methods, systems, challenges. Springer Nature, 2019

20. Mahesh B. Machine learning algorithms-a review. International Journal of Science and Research (IJSR).[Internet] 2020 ; 9 :381-6

21. Roscher R, Bohn B, Duarte MF et Garcke J. Explainable machine learning for scientific insights and discoveries. Ieee Access 2020 ; 8 :42200-16

22. Berry MW, Mohamed A et Yap BW. Supervised and unsupervised learning for data science. Springer, 2019

23. Govender P et Sivakumar V. Application of k-means and hierarchical clustering techniques for analysis of air pollution : A review (1980–2019). Atmospheric Pollution Research 2020 ; 11 :40-56

24. Ahmed M, Seraj R et Islam SMS. The k-means algorithm : A comprehensive survey and performance evaluation. Electronics 2020 ; 9 :1295

25. Khan K, Rehman SU, Aziz K, Fong S et Sarasvady S. DBSCAN : Past, present and future. *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*. IEEE. 2014 :232-8

26. Wani MA, Bhat FA, Afzal S et Khan AI. Advances in deep learning. Springer, 2020

27. Lee AX, Nagabandi A, Abbeel P et Levine S. Stochastic latent actor-critic : Deep reinforcement learning with a latent variable model. Advances in Neural Information Processing Systems 2020 ; 33 :741-52

28. Cardoso JF. Blind signal separation : statistical principles. Proceedings of the IEEE 1998 ; 86 :2009-25

29. Edgar RC et Batzoglou S. Multiple sequence alignment. Current opinion in structural biology 2006 ; 16 :368-73

30. Aniba MR, Poch O et Thompson JD. Issues in bioinformatics benchmarking : the case study of multiple sequence alignment. Nucleic acids research 2010 ; 38 :7353-63

31. Doolittle RF, Feng DF, Johnson M et McClure M. Origins and evolutionary relationships of retroviruses. The Quarterly Review of Biology 1989 ; 64 :1-30

32. Larkin MA, Blackshields G, Brown NP, Chenna R, McGettigan PA, McWilliam H, Valentin F, Wallace IM, Wilm A, Lopez R et al. Clustal W and Clustal X version 2.0. bioinformatics 2007 ; 23 :2947-8

33. Simonsen M, Mailund T et Pedersen CN. Rapid neighbour-joining. *International Workshop on Algorithms in Bioinformatics*. Springer. 2008 :113-22

34. Irie K, Gotoh Y, Yashar BM, Errede B, Nishida E et Matsumoto K. Stimulatory effects of yeast and mammalian 14-3-3 proteins on the Raf protein kinase. Science 1994 ; 265 :1716-9

35. Edgar RC. MUSCLE : multiple sequence alignment with high accuracy and high throughput. Nucleic acids research 2004 ; 32 :1792-7
36. Morgenstern B. DIALIGN : multiple DNA and protein sequence alignment at BiBiServ. Nucleic acids research 2004 ; 32 :W33-W36
37. Subramanian AR, Kaufmann M et Morgenstern B. DIALIGN-TX : greedy and progressive approaches for segment-based multiple sequence alignment. Algorithms for Molecular Biology 2008 ; 3 :1-11
38. Kumar M. An enhanced algorithm for multiple sequence alignment of protein sequences using genetic algorithm. EXCLI journal 2015 ; 14 :1232
39. Li KB. ClustalW-MPI : ClustalW analysis using distributed and parallel computing. Bioinformatics 2003 ; 19 :1585-6
40. Zola J, Yang X, Rospondek A et Aluru S. T-Coffee : a parallel multiple sequence aligner. *PDCS*. 2007 :248-53
41. Nakamura T, Yamada KD, Tomii K et Katoh K. Parallelization of MAFFT for large-scale multiple sequence alignments. Bioinformatics 2018 ; 34 :2490-2
42. Vouzis PD et Sahinidis NV. GPU-BLAST : using graphics processors to accelerate protein sequence alignment. Bioinformatics 2011 ; 27 :182-8
43. Blazewicz J, Frohmberg W, Kierzynka M et Wojciechowski P. G-MSA—A GPU-based, fast and accurate algorithm for multiple sequence alignment. Journal of Parallel and Distributed Computing 2013 ; 73 :32-41
44. Liu Y, Cui H et Zhao R. Fast Acquisition of Spread Spectrum Signals Using Multiple GPUs. IEEE Transactions on Aerospace and Electronic Systems 2019 ; 55 :3117-25
45. Lebsir R, Layeb A et Fariza T. A Greedy Clustering Algorithm for Multiple Sequence Alignment. International Journal of Cognitive Informatics and Natural Intelligence (IJCINI) 2021 ; 15 :1-17
46. Zafalon G, Visotaky J, Amorim A, Valêncio C, Neves L, De Souza R et Machado J. A parallel approach of COFFEE objective function to multiple sequence alignment. *Journal of Physics : Conference Series*. T. 633. 1. IOP Publishing. 2015 :012084
47. Agarwal P et Rizvi S. Solving sequence alignment problem using pipeline approach. Bharati Vidyapeeth's Institute of Computer Applications and Management 2009 ; 107
48. Huang S, Holt J, Kao CY, McMillan L et Wang W. A novel multi-alignment pipeline for high-throughput sequencing data. Database 2014 ; 2014
49. Naveed T, Siddiqui IS et Ahmed S. Parallel needleman-wunsch algorithm for grid. *Proceedings of the PAK-US International Symposium on High Capacity Optical Networks and Enabling Technologies (HONET 2005), Islamabad, Pakistan*. 2005
50. Dohi K, Benkridt K, Ling C, Hamada T et Shibata Y. Highly efficient mapping of the Smith-Waterman algorithm on CUDA-compatible GPUs. *ASAP 2010-21st IEEE International Conference on Application-specific Systems, Architectures and Processors*. IEEE. 2010 :29-36

51. Helal M, El-Gindy H, Mullin L et Gaeta B. Parallelizing optimal multiple sequence alignment by dynamic programming. *2008 IEEE International Symposium on Parallel and Distributed Processing with Applications*. IEEE. 2008 :669-74

52. Saeed F et Khokhar A. A domain decomposition strategy for alignment of multiple biological sequences on multiprocessor platforms. Journal of Parallel and Distributed Computing 2009 ; 69 :666-77

53. Edgar RC. Search and clustering orders of magnitude faster than BLAST. Bioinformatics 2010 ; 26 :2460-1

54. Li W et Godzik A. Cd-hit : a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics 2006 ; 22 :1658-9

55. Dondoshansky I et Wolf Y. Blastclust (ncbi software development toolkit). NCBI, Bethesda, Md 2002 ; 14

56. Bruneau M, Mottet T, Moulin S, Kerbiriou M, Chouly F, Chretien S et Guyeux C. A clustering package for nucleotide sequences using Laplacian Eigenmaps and Gaussian Mixture Model. Computers in Biology and Medicine 2018 ; 93 :66-74

57. Zhu X, Li K et Salah A. A data parallel strategy for aligning multiple biological sequences on multi-core computers. Computers in biology and medicine 2013 ; 43 :350-61

58. Tang T et Li J. Transformation of FASTA files into feature vectors for unsupervised compression of short reads databases. Journal of bioinformatics and computational biology 2021 ; 19 :2050048

59. Lebsir R, Layeb A et Abdaoui N. ACE-MSA : Application for Creating and Evaluating Multiple Sequence Alignment. International Conference on Innovative Trends in Computer Science (CITCS) 2019 ; 1 :1-17

60. Prasad DV, Madhusudanan S et Jaganathan S. uCLUST-A new algorithm for clustering unstructured data. ARPN Journal of Engineering and Applied Sciences 2015 ; 10 :2108-17

61. PRADO J. Introduction à MATLAB. Ed. Techniques Ingénieur, 2005

62. Thompson JD, Koehl P, Ripp R et Poch O. BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. Proteins : Structure, Function, and Bioinformatics 2005 ; 61 :127-36

63. Spark A. Apache spark. Retrieved January 2018 ; 17 :1