الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة الإخوة منتوري قسنطينة ا
Frères Mentouri Constantine I University
Université Frères Mentouri Constantine I

Université Frères Mentouri Constantine
Faculté des Sciences de la Nature et de la Vie
Département de biologie appliquée

جامعة الاخوة منتوري قسنطينة
كلية علوم الطبيعة والحياة
قسم البيولوجيا التطبيقية

Mémoire présenté en vue de l'obtention du diplôme de Master
Domaine : Sciences de la Nature et de la Vie
Filière : Sciences Biologiques
Spécialité : BIOINFORMATIQUE

N° d'ordre :
N° de série :

Intitulé

Optimisation de l'alignement multiple des séquences avec la métaheuristique de recherche coucou

---

Présenté par : **HAFIDI MOUHAMED SAMER** **Le 21/06/2022**
DIOUANE LOUAI MOHAMED AZIZ

Jury d'évaluation :

Encadreur : DR. GHERBOUDJ AMIRA (MCA- Université Frères Mentouri, Constantine1).

Examinateur 1 : DR. DAAS Mohamed Skander **(**MCA - Université Frères Mentouri, Constantine 1).

Examinateur 2 : DR**.** BELLIL Ines **(**MCA - Université Frères Mentouri, Constantine 1).

Année universitaire 2021 - 2022

Remercîment

EN PRÉAMBULE À CE MÉMOIRE NOUS REMERCIANT ALLAH QUI NOUS A AIDÉ ET NOUS A DONNÉ LA PATIENCE ET LE COURAGE DURANT CES LONGUES ANNÉES D'ÉTUDE. NOUS TENONS À REMERCIER PAR CE TRAVAIL TOUS CEUX QUI NOUS ONT AIDÉS DE LOIN OU DE PRÈS À RÉALISER CE MÉMOIRE DE FIN D'ÉTUDE

NOUS TENONS À REMERCIER SINCÈREMENT MADAME GHERBOUDJ AMIRA EN TANT QU'ENCADRANTE DE MÉMOIRE, ELLE A TOUJOURS MONTRÉ L'ÉCOUTE ET LA DISPONIBILITÉ TOUT AU LONG DE LA RÉALISATION DE CE MÉMOIRE, AINSI POUR L'INSPIRATION, L'AIDE ET LE TEMPS QU'ELLE A BIEN VOULU NOUS CONSACRER POUR QUE CE MÉMOIRE VOIT LE JOUR

N N'OUBLIE PAS NOS PARENTS POUR LEUR CONTRIBUTION, LEUR SOUTIEN ET LEUR PATIENCE AUSSI NOS FRÈRES ET NOS SŒURS QUI NOUS ONT TOUJOURS ENCOURAGÉS AU COURS DE LA RÉALISATION DE CE MÉMOIRE ET MES CHÈRES AMIES QUI M'ONT TOUJOURS AIDÉ SUR TOUS LES PLANS ET N'ONT PAS HÉSITÉS POUR FAIRE TOUS SES EFFORTS POUR MON BIEN.

POUR FINIR, JE REMERCIE MON ENTOURAGE POUR LE SOUTIEN APPORTÉ TOUT AU LONG DE CETTE ANNÉE D'ÉTUDE. MERCI ÉGALEMENT POUR LA RELECTURE DE MON MÉMOIRE. MERCI D'AVOIR PRIS LE TEMPS ET D'AVOIR EU LA PATIENCE DE LE FAIRE

Résumé

La bioinformatique est une discipline qui vise le traitement automatique de l'information biologique. L'alignement multiple de séquences (MSA) constitue une tâche fondamentale pour beaucoup d'applications en bio-informatique y compris : la prédiction des structures primaires et secondaires des séquences, la détection de la distance phylogénétique, la prédiction des structures des molécules… etc.

Dans ce mémoire de fin d'étude, nous avons présenté les différentes méthodes d'alignement multiple des séquences. Ensuite, nous avons travaillé sur la métaheuristique nommée « Recherche Coucou » (en anglais : Cuckoo Search 'CS'). Pour cela, nous avons construit des fonctions pour adapter et utiliser l'algorithme CS pour l'alignement multiple des séquences. Les résultats obtenus ont été comparés avec ceux d'autres méthodes présentées dans la littérature. Cette comparaison a montré l'efficacité de la méthode proposée.

الملخص :
المعلوماتية الحيوية هي تخصص يهدف إلى المعالجة التلقائية للمعلومات البيولوجية. تعد محاذاة التسلسل المتعدد (MSA) مهمة أساسية للعديد من تطبيقات المعلوماتية الحيوية بما في ذلك: التنبؤ بهياكل التسلسل الأولي والثانوي، واكتشاف مسافة النشوء والتطور، والتنبؤ بالتركيبات الجزيئية، وما إلى ذلك.
في هذه الرسالة، قدمنا الطرق المختلفة لمحاذاة التسلسل المتعدد. بعد ذلك، عملنا على طريقة تسمى" بحث كوكو) "بالإنجليزية .("Cuckoo Search "CS : لهذا، قمنا ببناء وظائف لتكييف واستخدام خوارزمية CS لمحاذاة التسلسل المتعدد. تمت مقارنة النتائج التي تم الحصول عليها مع تلك الخاصة بالطرق الأخرى المعروضة في المقالات. أظهرت هذه المقارنة فعالية الطريقة المقترحة.

Abstract

Bioinformatics is a discipline that aims at the automatic processing of biological information. Multiple sequence alignment (MSA) is a fundamental task for many bioinformatics applications including: primary and secondary sequence structure prediction, phylogenetic distance detection, molecular structure prediction...

In this thesis, we have presented the different methods of multiple sequence alignment. Then, we worked on the metaheuristic named "Cuckoo Search 'CS'". For this, we have built functions to adapt and use the CS algorithm for multiple sequence alignment. The results obtained were compared with those of other methods presented in the literature. This comparison showed the efficiency of the proposed method.

Liste des figures

Liste des tableaux

BFS	Best-First Search (BFS).
BrFS	Breadth-First Search.
CBFS	Cyclic Best-First Search.
COMPASS	Comparison of Multiple Protein Alignments with Assessment of Statistical Significance
CS	Coucou search
DCA	Divide and Conquer Algorithm.
DFS	Depth-First Search.
HMM	Hidden Markov Model
MSA	Multiple **S**équence **A**lignement.
PCMA	Profile Consistency Multiple Sequence Alignment.
PSO	Optimisation par esaim de particules
T-Coffee	Tree-based Consistency Objective Fonction for alignement Evaluation.
Tpop	Est la taille de la population

Table des matières

Table des matières

Introduction générale

La bioinformatique est un domaine pluridisciplinaire où l'informatique joue un rôle prépondérant. C'est une science qui conceptualise la biologie en termes de molécules et applique des " techniques d'informatiques" pour modéliser, analyser, comparer et simuler l'information biologique incluant séquences, structures, fonctions et phylogénie.

L'alignement multiple de séquences ou MSA (pour **M**ultiple **S**équence **A**lignement) est un problème fondamental en biologie moléculaire et représente une tâche de base pour beaucoup d'applications en bioinformatique. Il vise à apparier au sens biologique plusieurs séquences nucléiques et protéiques. Le MSA est le moyen utilisé par les biologistes pour analyser des séquences d'ADN (nucléiques) ou de protéines (protéiques) afin de déterminer leur degré d'homologie ou de divergence.

Needdlman et Wansch sont les pionniers dans la résolution du problème MSA. Ils ont proposé en 1970 une méthode basée sur la programmation dynamique. Il s'agit d'une méthode exacte qui fournit le résultat optimal.

Cependant, La recherche d'un alignement de bonne qualité implique souvent une exploration de l'espaces de recherche très vaste et dont la complexité devient de plus en plus critique avec le nombre et les taille des séquences à aligner. Cependant trouver un alignement multiple a été démontré comme un problème NP-complet. Raison pour laquelle, le MSA ne peut être résolu par une méthode exacte que pour des séquences de petites tailles et dont le nombre est réduit induisant des espaces de tailles réduites.

Par conséquent, plusieurs d'autres méthodes ont été proposées dans la littérature pour minimiser les coûts de résolution du problème d'MSA. Ces méthodes sont des méthodes approchées qui permettent de proposer des alignement semi optimaux voir optimaux avec des coûts de réponse raisonnables. Parmi ces méthodes on peut citer : les méthodes progressives comme la méthode CLUSTAW [1], les méthodes itératives comme la métaheuristique MAAFT [2], les méthodes basées sur la consistance comme la méthode PCMA [3] et les méthodes évolutionnaires comme la métaheuristique dites algorithme génétique [4].

Dans ce mémoire, nous présentons une utilisation avec adaptation de la métaheuristique de Recherche Coucou (en anglais : Cuckoo Search 'CS') pour résolution du problème MAS. CS est une des méthodes d'optimisation qui s'inspire de l'intelligence par essaim, une catégorie alternative de la catégorie des algorithmes évolutionnaires.

Notre mémoire comme suit :

Le chapitre 1 : Nous abordons des notions en relation avec la biologie moléculaire y compris séquençage de l'ADN, prédiction de structures, l'alignement des séquences et l'assemblage des séquences.

Le chapitre 2 : Nous présentons les méthodes d'alignement de séquences les plus connues et utilisées dans la littérature.

Le chapitre 3 : Nous présentons notre méthode avec l'étude expérimentale réalisée et les résultats obtenus.

CHAPITRE I :

Biologies Moléculaire

I .1. Introduction

La bio-informatique est l'utilisation des technologies de l'information dans le domaine de la biologie moléculaire. Bio-informatique implique maintenant la création et le développement de base de données, des algorithmes, des techniques informatiques et statistiques et de la théorie pour résoudre les problèmes formels et pratiques découlant de la gestion et l'analyse des données biologiques. Dans ce chapitre, on propose une présentation générale sur les notions Primaires da la biologie moléculaire et la bio-informatique

I .2. Les définitions

I .2.1. Séquençage de l'ADN

I .2.1.1. Définition

Le séquençage de l'ADN consiste à déterminer la séquence nucléotidique d'un gène ou d'un fragment de gène [5], ce procédé repose sur la synthèse d'un brin d'ADN par une ADN polymérase, il a été mis au point pour la première fois en **1977** [6].

La séquence d'ADN contient l'information nécessaire aux êtres vivants pour survivre et se reproduire. La détermination de cette séquence est utile pour les recherches visant à savoir comment vivent les organismes que pour des sujets appliqués.
En médecine, elle peut être utilisée pour identifier, diagnostiquer et potentiellement
Trouver des traitements à des maladies génétiques.

En biologie, l'étude des séquences d'ADN est devenue un outil important pour la classification des espèces.

I .2.1.2. Etapes du séquençage

- Le séquençage se passe dans un tube à essai en présence des acteurs de la synthèse d'ADN :
    - ADN à séquencer,
    - Nucléotides,
    - Amorce,
    - ADN polymérase.

- ❖ L'ADN polymérase utilise aléatoirement les nucléotides présents dans le milieu pour copier le brin matrice en synthétisant un ADN de séquence complémentaire.
- ❖ Lorsque l'ADN polymérase choisit par hasard un didésoxynucléotide (ce qui est rare puisqu'il y en a moins que des nucléotides) et qu'elle l'incorpore dans la chaîne en synthèse, celle-ci s'interrompt prématurément. Chaque didésoxynucléotide est marqué par un fluorochrome différent (A vert, T rouge, G jaune et C bleu), une chaîne qui se termine par exemple par un A sera verte.
- ❖ Puisqu'un grand nombre de réactions de synthèse ont lieu dans le tube, il existe Statistiquement des chaînes de toutes les tailles (correspondant à un arrêt de la synthèse à chaque nucléotide) et beaucoup de fragments d'une même taille. Ces chaînes commencent toutes au même endroit sur l'ADN matrice, toutes celles qui possèdent la même longueur se terminent par le même didésoxynucléotide marqué.
- ❖ Il est alors possible de séparer les chaînes d'ADN obtenues en fonction de leur taille sur un gel d'acrylamide en présence d'un courant électrique.

Plus les chaînes sont courtes, plus elles migrent loin et tous les fragments d'une même Taille migrent à la même distance. On obtient alors une succession de bandes colorées, chacune correspondant au dernier nucléotide incorporé. Il suffit de lire la succession des couleurs pour connaître l'ordre des nucléotides.

I .2.1.3. Les techniques de séquençage

- ➢ Technique de Maxam-Gilbert : Cette technique est une méthode chimique de traitement de l'ADN. Un fragment amplifié par PCR et marqué radioactivement par le phosphore radioactif (P32) est modifié par un agent chimique, par exemple l'hydralazine. Celle-ci modifie les bases C et T et en milieu alcalin, uniquement les bases C (comme dans ce schéma). Dans un second temps, l'addition de pipéridine casse de manière aléatoire et au moins une fois au niveau de chaque base C modifié. On obtient donc des fragments de taille différente.

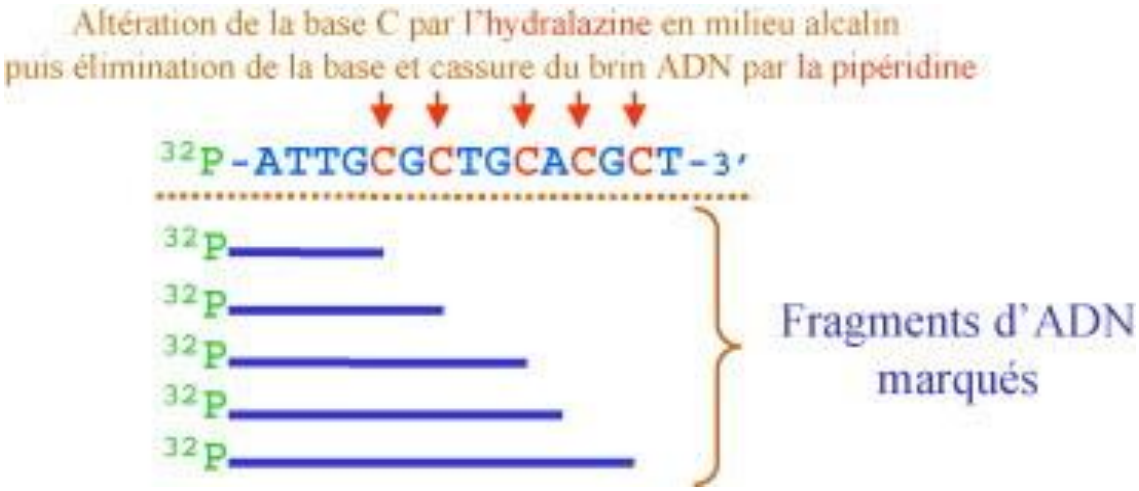

Figure 1 : Technique de Maxam-Gilbert [57]

- **Séquençage selon la méthode de Sanger**

Après dénaturation du produit amplifié par séquençage, l'un des deux brins (ici, le brin sens) s'hybride à une amorce spécifique. Pour la simplicité du schéma, nous avons pris une amorce de 5 pb, la taille habituelle des amorces étant de 20 pb environ. Le mélange réactionnel contient, outre les tampons et l'ADN polymérase, des déoxynucléotides triphosphates (dNTP, dA-, dC-, dG-, dT-TP) mais aussi des didéoxynucléotides triphosphates (ddNTP, Duda-, ddC-, ddG-, ddT-TP). L'incorporation aléatoire d'un ddNTP à la place d'un dNTP ne permet plus la polymérisation par l'ADN polymérase. L'extension s'arrête. À la fin de la réaction de séquence effectuée selon des cycles thermiques identiques à ceux de la PCR (on parle de PCR asymétrique, une seule amorce étant utilisée au lieu de deux), nous avons des fragments de taille différente. Ces fragments sont soumis à migration dans un champs électrique. Il s'agit le plus souvent d'une électrophorèse capillaire. Chaque ddNTP étant marqué par un fluorophore différent, un signal lumineux sera généré, spécifique de la base didéoxy incorporée. Les fragments étant de taille différente et la résolution allant jusqu'à une base de différence, il sera simple de recueillir ce signal et en déduire la séquence. Les signaux lumineux sont analysés par un logiciel spécifique, et le résultat de l'analyse peut être lu, par exemple, sous forme d'un électrophorégramme de lecture facile. Des logiciels d'interprétation des séquences sont également disponibles. Pour confirmer un résultat, toute réaction de séquence d'un fragment d'ADN est systématiquement faite sur le brin sens et le brin antisens.

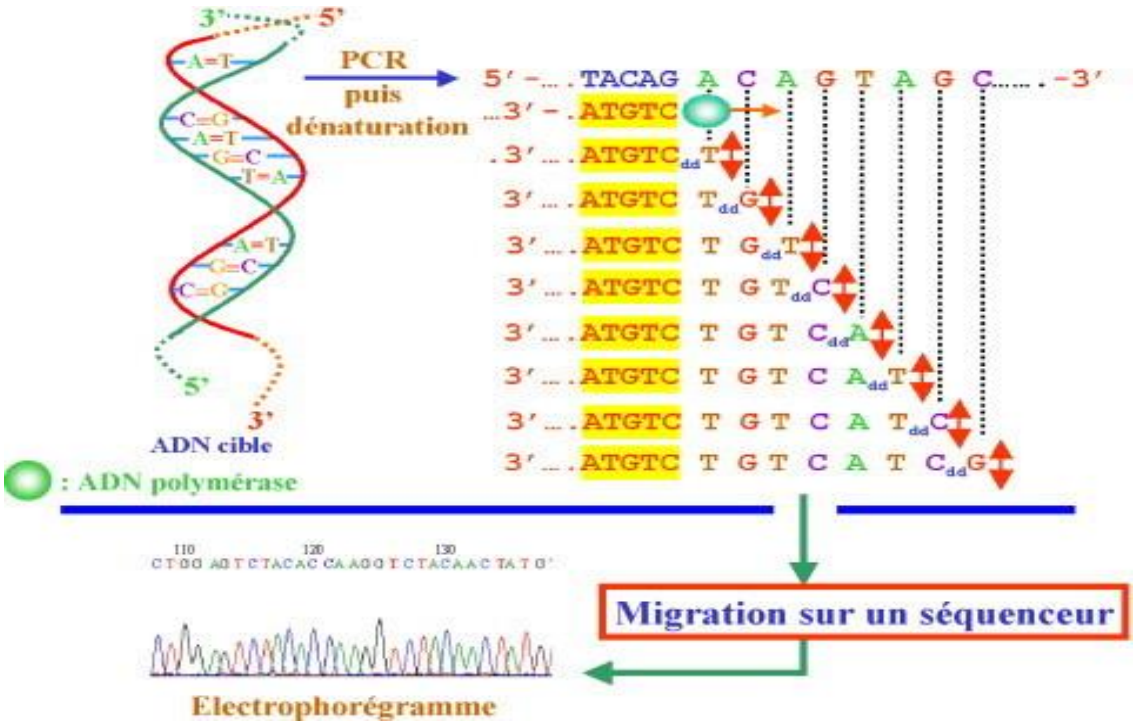

Figure 2 : la méthode de Sanger [58]

I .2.1.4. Intérêt général du séquençage

En **2001**, 95 % de l'ADN humain était séquencé et en **2006** il a été finalisé [7,8].Par ailleurs, les génomes de nombreux agents infectieux de mammifères et de plantes Ont également été séquencés dans leur totalité, leur connaissance a modifié considérablement les recherches biomédicales et biologiques en ouvrant de vastes panoramas dans le domaine de la médecine (diagnostic, thérapeutique, prédiction, pronostic, prévention. . .) et dans de nombreuses autres disciplines biologiques (anthropologie, agronomie, environnement. . .).

Les étapes suivantes représentent quelques applications de séquençage :

- Diagnostic et traitement de nombreuses maladies humaines (exemples : cancers, maladies infectieuses, maladies héréditaires…)
- Informations sur le génome (structure, fonction, évolution) et étude des variations du génome (polymorphismes bi alléliques, insertions, délétions, insertions/délétions (appelées aussi indes), réarrangement de gènes, variation du nombre de copies de gènes, duplication (ou plus)

- Variants génétiques associées à une pathologie (par exemple, le diabète)
- Analyse de méthylation du génome (études épigénétiques et méthylome)
- Analyse microbiologique (identification d'espèces, taxonomie, études épidémiologiques, génotypage à but pronostique et/ou thérapeutique)
- Tests de paternité et médecine légale
- Police scientifique
- Pharmacogénétique
- Études anthropologiques

I .2.2. Prédiction de structures

Le nombre de structures primaires possibles pour les protéines est exponentiel en fonction de la longueur l. En revanche le nombre de combinaisons de structures tridimensionnelles est beaucoup plus réduit. Ainsi, des séquences très différentes peuvent avoir des structures similaires. La structure tridimensionnelle d'une séquence est une information contenue dans sa structure primaire. Cependant, cette information est actuellement difficile à déterminer [9]. Sans utiliser une analyse directe de la séquence.

Connaitre la structure primaire d'une protéine ne permet pas actuellement d'en déduire sa structure tridimensionnelle.
La structure 3D d'une protéine peut être déterminée expérimentalement par cristallographie ou par résonance magnétique nucléaire. Ces méthodes sont toutefois assez lourdes à mettre en œuvre, et nécessitent un matériel spécialisé. La prédiction de structures est une branche de la bio-informatique qui consiste à essayer de déterminer la structure d'une protéine sans passer par la phase expérimentale.
La méthode qui donne les meilleurs résultats actuellement procède par homologie,

C'est-`a-dire en se basant sur des séquences ayant des structures primaires assez proches, et dont la structure tridimensionnelle est déjà connue. Il s'agit l`a d'une application directe du problème d'alignement de séquences. Il est nécessaire pour cela de trouver des séquences similaires.

I .2.3. L'alignement des séquences

En bio-informatique , un alignement de séquences est une façon d'organiser l' des séquences primaires de l'ADN , ARN, ou protéine pour identifier des régions de similarité qui peuvent être une conséquence de fonctionnelle, structurelles ou évolutives relations entre les séquences[10]. Séquences alignées de nucléotidiques ou d'acides aminés sont typiquement de résidus représentés sous forme de lignes dans une matrice . Les lacunes sont insérées entre les résidus de sorte que les résidus avec des caractères identiques ou similaires sont alignés en colonnes successives.

Si deux séquences dans un alignement partagent un ancêtre commun, décalages peuvent être interprétées comme des mutations ponctuelles et des lacunes que indels (c'est-à insertion ou la suppression des mutations) introduites dans une ou deux lignées dans le temps car ils divergent les uns des autres. Dans un alignement de séquences de protéines, le degré de similitude entre les acides aminés occupant une position particulière dans la séquence peut être interprété comme une mesure approximative de la conserver une région particulière ou motif de séquence est l'une des lignées. L'absence de substitutions, ou la présence de seulement des substitutions très conservatrices (c'est la substitution d'acides aminés dont chaînes latérales ont des propriétés biochimiques similaires) dans une région particulière de la séquence, suggèrent que cette région a une importance structurelle ou fonctionnelle. Bien que l'ADN et l'ARN bases nucléotidiques sont plus similaires les uns aux autres pour que les acides aminés, la conservation de l'appariement de base peut indiquer un rôle fonctionnelle ou structurelle similaire. L'alignement de séquences peut être utilisée pour les séquences non-biologiques, tels que ceux présents dans langage naturel ou dans les données financières [11].

Très court ou très séquences similaires peuvent être alignés à la main ; Cependant, les problèmes les plus intéressants nécessitent l'alignement de séquences longues, très variables ou extrêmement nombreux qui ne peuvent pas être alignées uniquement par l'effort humain. Au lieu de cela, la connaissance humaine est principalement appliquée pour la construction des algorithmes pour produire des alignements de séquences de haute qualité, et parfois à ajuster les résultats définitifs de refléter les habitudes qui sont difficiles à représenter algorithmique (surtout dans le cas de séquences nucléotidiques). Approches informatiques à l'alignement de séquence répartissent généralement en deux catégories : *les alignements mondiaux* et *alignements locaux.* Calcul d'un alignement global est une forme d'optimisation globale que les « forces » l'alignement s'étende sur toute la longueur de toutes les séquences de

la requête. En revanche, les alignements locaux identifient des régions de similitude dans les longues séquences qui sont souvent très divergentes globale. Alignements locaux sont souvent préférables, mais peut être plus difficile à calculer en raison de la difficulté supplémentaire d'identifier les régions de similarité. Une variété d'algorithmes de calcul a été appliquées au problème de l'alignement de séquence, y compris, mais lent formellement l'optimisation des méthodes telles que programmation dynamique et efficace heuristiques ou probabilistes méthodes conçues pour la recherche de base de données à grande échelle.

I .2.3.1. Principe de l'alignement

- **Aligner des séquences =** Rechercher le maximum d'appariements entre les résidus des séquences comparées. L'alignement est d'autant plus parfait qu'il n'y a pas de

Mésappariements et de brèches.

- **Mesure du degré de similitude :** La plupart des méthodes d'alignement de séquences

Biologiques, et en particulier les méthodes d'alignement de séquence de protéines cherchent à optimiser un score d'alignement. Ce score est relié au taux de similarité entre les deux séquences comparées.

I .2.3.2. Trois processus d'alignements : global, multiple et local

- **Alignement global :** Alignement de deux séquences sur la totalité de leur longueur en tenant compte de tous les résidus. Si les longueurs sont différentes, des insertions / délétions sont introduites pour aligner les deux extrémités des deux séquences.
  Cet alignement permet de mesurer le degré de similitude entre deux séquences

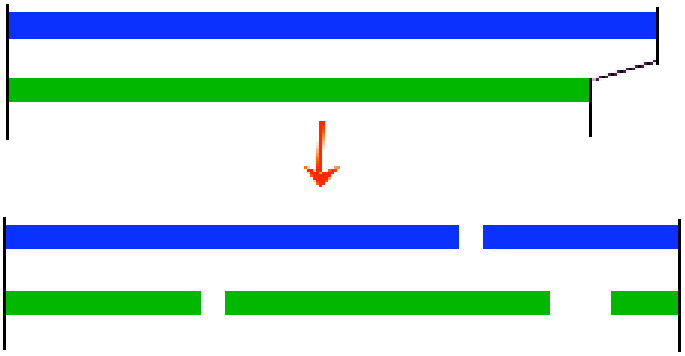

Figure 3 : Exemple d'alignement global

- **Alignement local :** Alignement de deux séquences sur des régions isolées et permettant de trouver des segments qui ont un haut degré de similitude. Utilisé pour la recherche dans les bases de données (comparaison d'une séquence avec les séquences contenues dans la base)

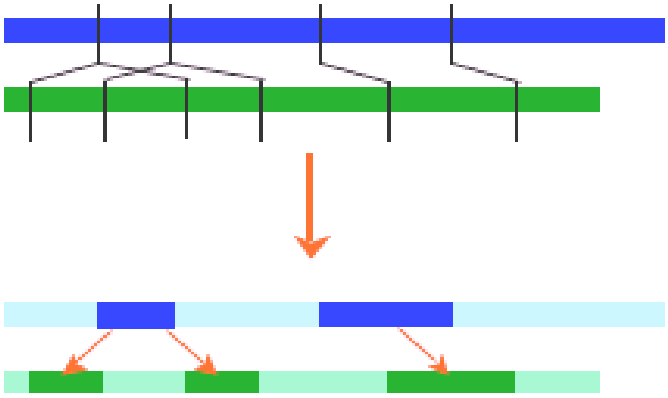

Figure 4 : Exemple d'alignement local

- **Alignement Multiple :** Alignement portant sur plusieurs séquences à la fois et dans leur intégralité. Il permet de mettre en évidence les relations entre séquences que l'on ne peut pas visualiser en comparant les séquences deux à deux.

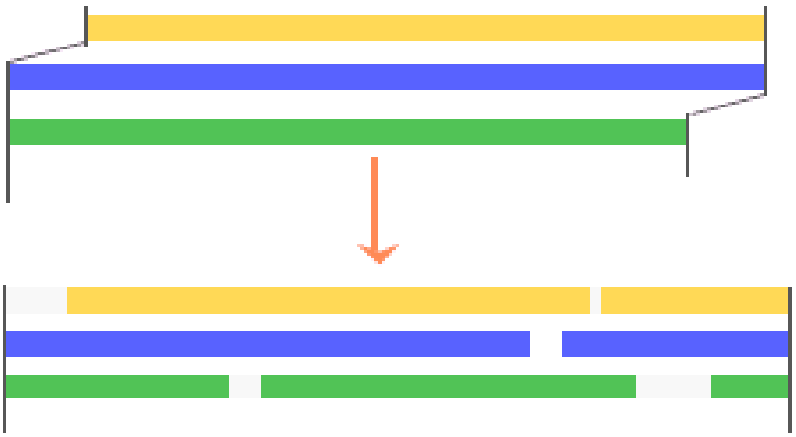

Figure 5 : Exemple d'alignement multiple

À chaque type d'alignement est associé un programme informatique permettant d'optimiser le Traitement

Tableau 1 : Les types des programmes de traitement d'alignement.

Alignement global :	**Alignement local :**	**Alignement Multiple :**

Needle Stretcher	BLAST (Basic Local Alignment Tool) FASTA	T-Coffee

I .2.3.3. Pourquoi aligner les séquences

L'alignement multiple de séquences MSA (Multiple Sequence Alignment) consiste à aligner plusieurs séquences dans leur intégralité afin de tirer les relations entre une famille de séquences. Le but principal de l'alignement multiple est de montrer les rapports essentiels et 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. Il permet aussi de construire la séquence consensus de plusieurs séquences alignées. Le MSA contribue efficacement à une meilleure compréhension de l'évolution des séquences biologiques. En plus, l'alignement multiple est également utilisé dans plusieurs autres domaines comme la bio-informatique structurelle.

Malheureusement, La construction manuelle d'un alignement multiple est une opération très fastidieuse et non praticable. Pour cela la construction automatique des alignements est devenue aujourd'hui une tâche importante en bio-informatique. Par ailleurs, le MSA est caractérisé par une grande complexité temporaire et spatiale.

I .2.4. Assemblage des séquences

L'assemblage des séquences est une étape cruciale dans le processus d'analyse des données [12]. Il consiste à reconstituer la séquence de l'ADN entier (que ce soit des gènes ou des génomes) à partir des lectures brutes (reads) produites par le séquenceur. Cette approche permet donc de découvrir de nouvelles séquences. Ce type de séquençage est essentiel pour la caractérisation de la biodiversité, mais l'assemblage de ces données demande une capacité informatique et des algorithmes plus complexes que la génomique comparative, ou le reséquençage de génomes. Il exige de plus la réalisation de nombreuses réactions de séquençage qui permettent de maximiser les chances d'obtenir les transcrits sur toute leur longueur. L'analyse du transcriptome.

D'organismes dont le génome n'est pas séquencé fait aussi appel à du séquençage/assemblage de novo, Ce type d'assemblage s'effectue sans référence génomique ou transcriptomique.

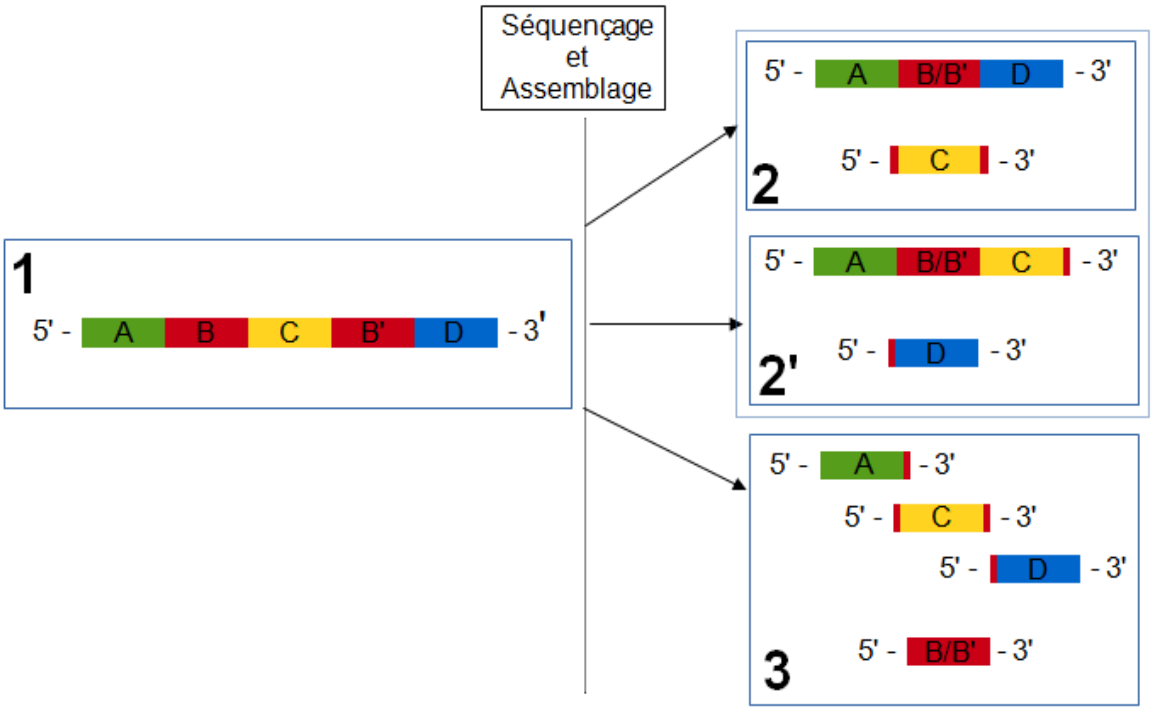

Figure 6 : L'assemblage des séquences

I .3. Conclusion

Dans ce chapitre nous avons présenté les domaines d'utilisation de la bioinformatique et quelques techniques et notions de bases concernant la biologie moléculaire. Dans le chapitre suivant, nous allons présenter les méthodes d'alignement multiple qui consistent à relier la séquence à la structure et à la fonction ainsi que caractériser les régions conservées et les régions variables.

CHAPITRE II :

Méthodes d’alignement multiple des séquences

II.1. Introduction

La résolution de différentes sortes de problèmes a poussé les chercheurs à proposer des méthodes de résolution et à réaliser de grands efforts pour améliorer leurs performances en termes de temps de calcul nécessaire et la qualité de la solution proposée

Dans ce chapitre, nous allons exposer les méthodes les plus récentes conçues pour résoudre le problème de MSA selon les approches utilisées.

II.2. Alignement Multiple de Séquences

L'alignement multiple des séquences d'ADN ou de protéines est une des techniques les plus utilisées dans l'analyse de séquence. Il est considéré parmi les problèmes les plus difficiles en bio-informatique.

L'alignement multiple de séquences (Multiple Séquence Alignement : MSA) est une tâche cruciale et très importante en biologie moléculaire. MSA offre aux biologistes un moyen pour analyser des séquences d'ADN ou de protéines et de déterminer par la suite leur degré d'homologie ou de divergence. *MSA* est utilisé dans la construction des arbres phylogénétiques et identifier les motifs dans des familles de protéines, ceci permet de prédire leur aspect structurel et fonctionnel.
La qualité d'une comparaison ou d'une prédiction dépend de la qualité du *MSA*.
Jusque récemment le choix d'une méthode pour la construction des alignements multiples de séquence (MSA) a été limité à une poignée de packages mais une augmentation récente des données génomique a poussée l'élaboration de plusieurs nouvelles méthodes, plus précises et plus rapides que les anciennes. Dans la pratique, ce large choix a également rendu difficile le choix objectif de la méthode appropriée pour un problème spécifique.

Pendant la dernière décennie, plus de 50 méthodes ont été décrites dans ce domaine et 20 uniquement pendant l'année 2005 [13]. Ce nombre risque d'augmenter car aucune parmi elles n'est totalement efficace pour tout type de séquences.

II.2. 1. Les Utilisation en informatique

L'alignement multiple de séquences permet de mettre en évidence les similarités entre plusieurs séquences. Il est donc possible de comparer simultanément la proximité de toutes ces séquences. Les informations apportées par ces comparaisons permettent d'obtenir des renseignements

importants sur les séquences comme les distances d'une séquence par rapport aux autres ou encore la mise en évidence de zones identiques entre plusieurs ou toutes les séquences.

II.3. Méthodes d'Alignements Multiple de Séquences

Dans la littérature, on rencontre trois catégories essentielles ou approches suivies pour construire un MSA. Néanmoins, ces approches sont parfois fusionnées, concaténées ou/et associées pour construire une seule méthode [14].

On distingue l'approche Exacte qui tente de donner plus de longévité à la programmation dynamique dans ce domaine et de déterminer un alignement optimal proprement dit comme elle le fait pour aligner deux séquences. De l'autre côté, on rencontre des heuristiques qui à leur tour se bifurquent en deux approches : Progressive et Itérative.
Les méthodes qui suivent l'approche progressive, sont reconnues d'être très rapides [15] et donnent des résultats assez satisfaisants mais leur inconvénient est le fait de s'arrêter sur les minima locaux et si une erreur est commise au début de l'alignement, elle va se propager sur l'alignement final.

L'approche itérative est une manière très simple, rapide et efficace permettant d'améliorer des méthodes d'alignement multiples. L'itération peut être employée pour améliorer le résultat d'un logiciel existant avec n'importe quelle fonction objective. Elle peut également être incorporée à une stratégie progressive d'alignement pour établir des alignements à partir de zéro pour produire encore de meilleurs résultats [16].

II.3.1. Méthodes Exactes

L'approche exacte n'est autre qu'une généralisation des méthodes de programmation dynamique [17]

La méthode de programmation dynamique utilisée pour aligner deux séquences, a été appliquée à l'alignement de plusieurs séquences (N dimensions) tels que MSA DCA [18]

Ce type de méthodes représente de gros problèmes : Le temps de calcul et l'espace mémoire.
· Dans la pratique, un alignement devient délicat pour un nombre de séquence N > 3, et même impossible pour N = 10
· Pour N séquences de longueur L, l'alignement optimal (au sens mathématique) nécessite :
- Un temps de calcul proportionnel à 2n Ln.

- Un espace mémoire proportionnel à Ln.

· Exemple : pour 10 séquences de 100 résidus, et 10-9 secondes de temps de calcul par colonne, nécessite alors :

Temps total = 210 * 10010 * 10-9 » 1014 s (> 3 106 années)

Espace mémoire = 1011 GB.

Le problème de l'alignement multiple exacte a été démontré être un problème *NP-complet*. D'où le recours aux méthodes approchées ou heuristiques.

II.3.1.1. Méthode Branch-and-Bound (B&B)

Le cadre Branch-and-Bound (B&B) est une méthodologie fondamentale et largement utilisée pour produire des solutions exactes aux problèmes d'optimisation NP-difficiles. La technique, qui a été proposée pour la première fois par Land et Doig [19], est souvent appelée algorithme ; cependant, il est peut-être plus approprié de dire que B&B encapsule une famille d'algorithmes qui partagent tous une procédure de solution de base commune. Cette procédure énumère implicitement toutes les solutions possibles au problème considéré, en stockant des solutions partielles appelées sous-problèmes dans une structure arborescente. Les nœuds inexplorés dans l'arborescence génèrent des enfants en partitionnant l'espace de solution en régions plus petites qui peuvent être résolues de manière récursive (c'est-à-dire, la ramification), et les règles sont utilisées pour élaguer les régions de l'espace de recherche qui sont manifestement sous-optimales (c'est-à-dire, les limites). Une fois que l'arbre entier a été exploré, la meilleure solution trouvée dans la recherche est renvoyée. Un premier aperçu de l'algorithme B&B de base a été fourni par Lawler et Wood [20] ; la procédure de résolution est également couverte dans les excellents textes de Nemhauser et Wolsey [21], Bertsimas et Tsitsiklis [22], et Papadimitriou et Steiglitz [23].

Cependant, dans le cadre ci-dessus, il y a trois composants qui ne sont pas spécifiés, mais qui peuvent avoir des impacts significatifs sur les performances de l'algorithme. Ces composants sont la stratégie de recherche (c. qui empêche l'exploration des régions sous-optimales de l'arbre). Clausen [24] donne un aperçu de ces différents composants et comment ils affectent les performances de l'algorithme pour le problème du voyageur de commerce, le problème de partitionnement de graphes et le problème d'affectation quadratique.

Des recherches substantielles ont été menées pour développer des extensions algorithmiques à la fois générales et spécifiques à ces trois composants qui peuvent améliorer les performances

de B&B. L'objectif est donc de fournir un aperçu des avancées modernes dans la théorie des algorithmes B&B, en particulier en ce qui concerne les trois composants ci-dessus. En outre, trois axes de recherche importants sont mis en évidence qui sont actuellement non résolus ou non étudiés dans la littérature. Ces axes de recherche sont :

- La formulation de nouvelles stratégies de recherche pour conduire rapidement un algorithme B&B vers une solution optimale.
- Une analyse de la façon dont la stratégie de branchement affecte à la fois la phase de recherche et de vérification, ainsi qu'une étude des nouvelles stratégies de branchement.
- Le développement d'une théorie unifiée pour l'élagage dans les algorithmes B&B, qui permettra de prouver de meilleures bornes théoriques sur les performances de ces algorithmes.

Relations entre les composants de l'algorithme

Il y a deux phases importantes dans tout algorithme B&B : la première est la phase de recherche, dans laquelle l'algorithme n'a pas encore trouvé de solution optimale x. La seconde est la phase de vérification, dans laquelle la solution en place est optimale, mais il reste encore des sous-problèmes inexplorés dans l'arbre qui ne peuvent pas être élagués. Notez qu'une solution existante ne peut pas être prouvée optimale tant qu'il ne reste plus de sous-problèmes inexplorés ; notez également que la délimitation entre la phase de recherche et la phase de vérification n'est pas connue tant que l'algorithme n'est pas terminé. Par un léger abus de terminologie, un problème P est dit résolu si l'algorithme B&B a terminé la phase de vérification. Dans ce cas, on dit que l'algorithme a produit un certificat d'optimalité. Les trois composants algorithmiques (stratégie de recherche, stratégie de branchement et règles d'élagage) jouent chacun un rôle distinct dans les algorithmes B&B par rapport à ces deux phases de fonctionnement a (Figure 1). En particulier, le choix de la stratégie de recherche impacte principalement la phase de recherche. Pour voir cela, supposons que les règles d'élagage ne dépendent que de la valeur de la solution en place (par exemple, elles comparent la borne inférieure d'un sous-problème à la valeur en place). Dans ce contexte, toute stratégie de recherche doit explorer le même ensemble de sous-problèmes une fois qu'une solution optimale est trouvée.

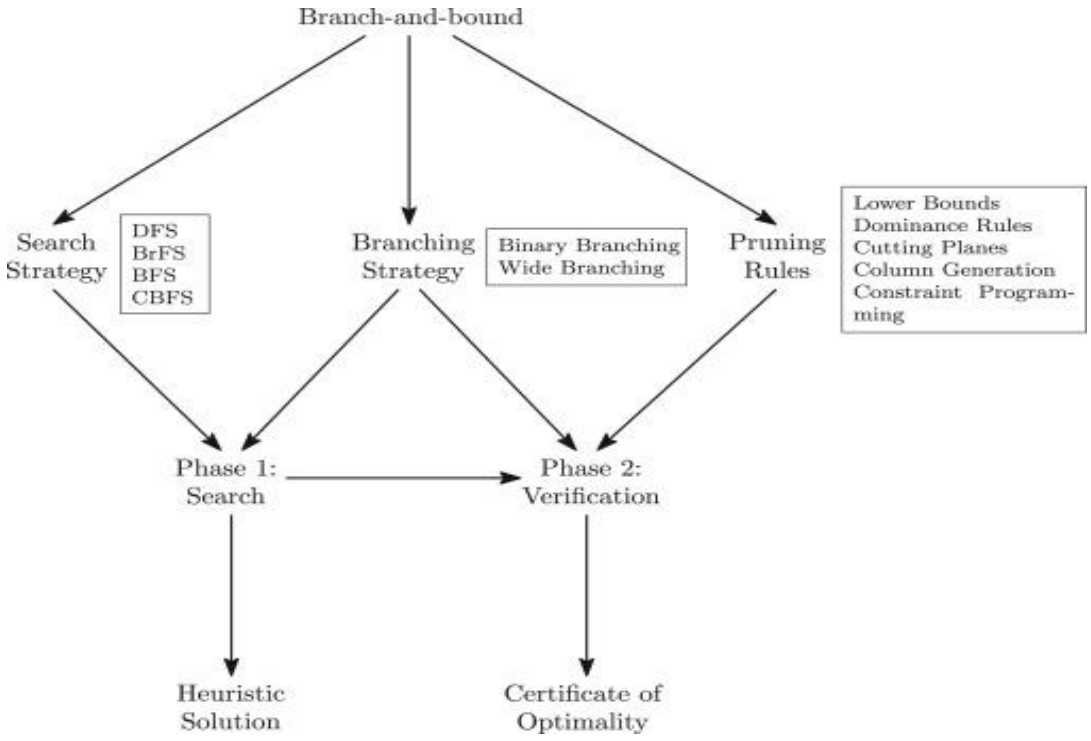

Figure 7 : Un schéma des trois principaux composants du B&B.

La stratégie de recherche et les règles d'élagage ont principalement un impact sur la phase de recherche et la phase de vérification, respectivement, tandis que la stratégie de branchement a un impact sur les deux.

D'autre part, les règles d'élagage sont souvent ciblées sur la phase de vérification, en particulier dans le cas d'une limitation basée sur les objectifs qui peut être relativement faible avant qu'une solution optimale (ou quasi optimale) ne soit connue. Dans ce cas, si la solution en place a une faible valeur objective au début du processus de recherche, les bornes inférieures ne pourront pas être élaguées efficacement, même si elles sont très serrées. Cependant, il existe également des situations dans lesquelles les règles d'élagage contribuent à la phase de recherche, comme lorsque des plans de coupe dans un programme mixte à nombres entiers (MIP) sont utilisés pour identifier des solutions réalisables.

Le troisième composant B&B, la stratégie de branchement, a des impacts significatifs à la fois sur la phase de recherche et sur la phase de vérification. En se ramifiant de manière appropriée aux sous-problèmes, la stratégie peut guider l'algorithme vers des solutions optimales. Une fois la phase de recherche terminée, une stratégie de branchement appropriée peut aider à limiter les décisions de branchement qui sont prises afin d'éviter qu'un travail inutile ne soit effectué pour produire un certificat d'optimalité.

Il existe deux raisons importantes pour améliorer les performances de l'algorithme B&B pendant la phase de recherche. Tout d'abord, si l'algorithme se termine avant de produire un certificat d'optimalité, la solution en place peut toujours être renvoyée comme une solution heuristique, ce qui peut être suffisant dans certains problèmes. Un exemple de ce comportement peut être vu dans [25 où B&B est utilisé pour améliorer les bornes supérieures pour les grandes instances d'équilibrage de la ligne d'assemblage simple même si un certificat d'optimalité ne peut pas être obtenu. Une autre approche de Guzelsoy et al. [26] appelé restrict-and-relax permet à B&B d'assouplir les décisions de branchement prises précédemment afin de trouver plus rapidement une bonne solution réalisable.

Deuxièmement, trouver une solution optimale plus tôt dans la phase de recherche a un impact direct sur la taille de l'arbre de recherche (et donc sur le temps nécessaire pour vérifier l'optimalité), car aucun autre nœud avec des bornes supérieures à la valeur optimale n'a besoin d'être exploré. Intuitivement, c'est la raison d'être du résultat de Dechter et Pearl [27] montrant que la meilleure recherche en premier explore le plus petit nombre de sous-problèmes de toute stratégie de recherche.

Cependant, il existe relativement peu de résultats récents dans la littérature étudiant les impacts de la stratégie de recherche et de la stratégie de branchement sur les performances des algorithmes B&B. Au lieu de cela, la plupart des travaux se concentrent sur les règles d'élagage, qui sont les plus utiles lors de la vérification. Le développement de stratégies de recherche plus avancées et de stratégies de branchement sont donc deux axes de recherche importants mis en évidence par cette enquête.

La figure 8 montre les relations internes entre les différents types de règles d'élagage, de stratégies de branchement et de stratégies de recherche. Dans cette figure, les lignes pleines indiquent une relation de généralisation. Par exemple, comme discuté dans la section 5.5, de nombreuses techniques de programmation par contraintes généralisent les plans coupants et les relations de dominance. De plus, la stratégie Cyclic Best-First Search (CBFS) est une généralisation de Depth-First Search (DFS), Breadth-First Search (BrFS) et Best-First Search (BFS). Les techniques de génération de colonnes, bien qu'elles ne soient pas strictement une généralisation d'autres techniques, sont étroitement liées aux techniques de limitation inférieure et de plan de coupe (en substance, la génération de colonnes ajoute des plans de coupe au problème d'optimisation double pour améliorer la limite inférieure calculée). Ce schéma met également en évidence la troisième recherche importante direction présentée dans cette enquête, à savoir l'absence d'une théorie généralisante forte des nombreux types de règles d'élagage utilisées dans les algorithmes B&B.

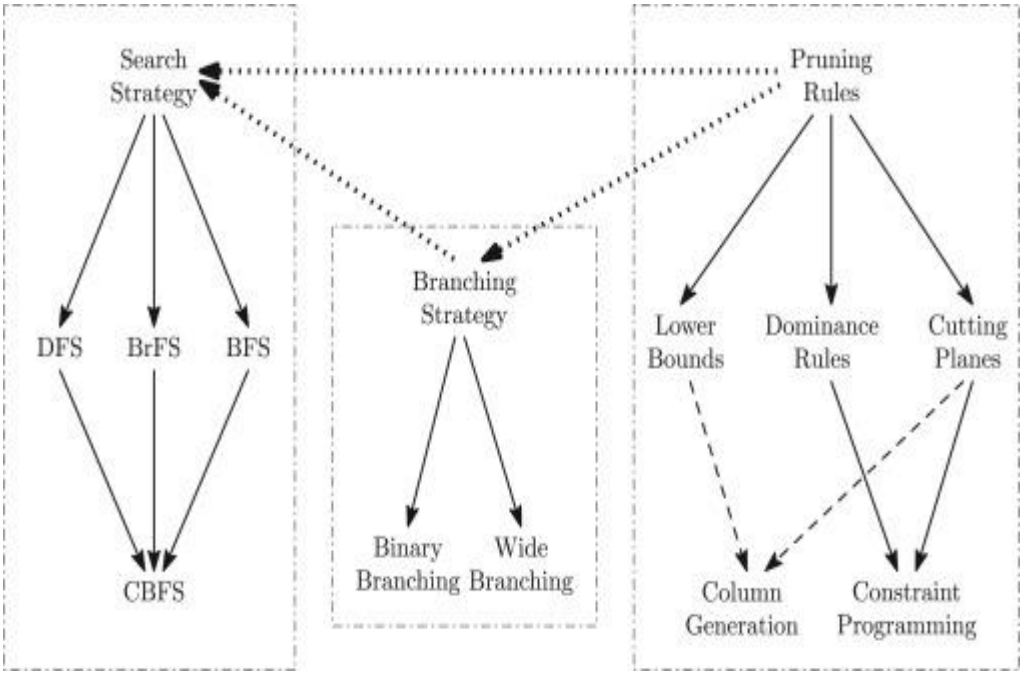

Figure 8 Diagram of Relationship between various algorithm components.

La Figure 8 montre également les relations entre les règles d'élagage, la stratégie de branchement et la stratégie de recherche utilisée par un algorithme. En particulier, le choix des règles d'élagage impacte ou limite souvent les choix qui peuvent être faits dans les deux autres domaines. Par exemple si la génération de colonnes est utilisée pour améliorer les bornes inférieures, le choix des stratégies de branchement qui peuvent être utilisées est limité. De plus, si des relations de dominance sont utilisées, cela peut faire de BrFS une stratégie de recherche souhaitable, car elle a la propriété de ne jamais explorer un sous problème dominé. Enfin, le choix de la stratégie de branchement peut lui-même impacter le choix de la stratégie de recherche. Par exemple, si la stratégie de branchement choisie produit un arbre particulièrement déséquilibré, la stratégie CBFS peut équilibrer le processus de recherche, ou des variantes de DFS peuvent limiter la profondeur explorée à n'importe quelle étape de l'algorithme.

II.3.1.2. Méthode MSA

C'est une tentative de rendre les algorithmes de la programmation dynamique conçus pour aligner deux séquences, opérationnels pour un alignement multiple. Sachant que la complexité temporelle et spatiale augmente proportionnellement avec le nombre et la longueur des séquences à aligner. Même la matrice de score devient elle aussi multidimensionnelle.

Le programme de MSA emploie un algorithme intelligent pour réduire le volume de la matrice de la programmation dynamique multidimensionnelle. L'algorithme de Carrillo et de Lipman était mis en application dans MSA.

Le score d'un alignement multiple généré par une heuristique est la somme des scores de tous alignements deux à deux définis pour l'alignement multiple. Sachant que :

- Le score *SP* pour toute paire de séquences extraite de l'alignement multiple optimal, devrait être inférieur au score *SP* optimal de l'alignement de paires de séquences.
- Le score *SP* total d'un alignement optimal devrait être plus grand que celui d'un alignement obtenu par des méthodes heuristiques.

En plaçant la limite inférieure et la limite supérieure, seulement un espace restreint doit être exploré dans la table de score multidimensionnelle [28]. Toutes ces considérations ont participé à la réduction du temps de calcul d'une manière significative.

Ce type de méthodes représente de gros problèmes : Le temps de calcul et l'espace mémoire.

Dans la pratique, un alignement devient délicat pour un nombre de séquence N > , et même impossible pour N = 10

Pour N séquences de longueur L, l'alignement optimal (au sens mathématique) nécessite :

-Un temps de calcul proportionnel à 2n Ln.

-Un espace mémoire proportionnel à Ln.

Le problème de l'alignement multiple exacte a été démontré être un problème NP-complet. D'où Le recours aux méthodes approche ou heuristiques.

II.3.1.3. Méthode de DCA

DCA (Divide and Conquer Algorithm) [29], c'est une heuristique basée sur l'idée ≪ diviser puis conquérir ». Le principe consiste à découper les séquences à aligner en sous-ensembles de segments. Ces segments doivent avoir une taille assez petite pour faciliter leur traitement par MSA. Les sous alignements produits sont alors concaténés pour former un seul alignement multiple final.

Comme étant une méthode exacte, elle hérite des mêmes inonvénients des méthodes de ce type : la complexité temporelle et spatiale.

II.3.2 Méthode Itératives

L'approche itérative a été employée plusieurs fois comme méthode d'optimisation pour produire des alignements multiples. Parfois elle est utilisée seule ou en combinaison avec d'autres méthodes. L'itération a un grand avantage parce qu'elle est souvent très simple soit en termes de code des algorithmes, soit en termes de complexité temporelle et spatiale.

Les étapes d'un alignement itératif :

· Repérer les deux séquences avec la plus forte similarité et les aligner avec une méthode de programmation dynamique.

· Trouver la séquence qui est la plus proche du profil obtenu avec les 2 séquences précédentes et l'aligner avec les deux autres par une méthode d'alignement profil-séquence.

- Répéter ceci jusqu'à ce que toutes les N séquences soient inclues dans l'alignement multiple

· Enlever la séquence S1 et la réaligner avec le profil obtenu avec les séquences de S2…Sn

- Répéter ceci pour toutes les autres séquences de S2 à Sn.

· Répéter l'étape précédente un certain nombre de fois ou arrêter le processus à convergence du score de l'alignement.

II.3.2.1. Méthode MAFFT

MAFFT [30] est un nouveau programme pour le problème de MSA. Il exploite les caractéristiques physico-chimiques des acides aminés qui composent les protéines pour établir le degré de similitude ou de divergence entre elles.

Une fois les valeurs de ces caractéristiques sont obtenues on applique une transformation de Fourier pour déterminer des relations entre les séquences à aligner afin de pouvoir générer un arbre guide comme toute méthode progressive le fait.

MAFFT a introduit des nouvelles techniques telles que :

1) Les régions homologues sont rapidement identifiées par l'exploitation de la transformation de Fourier (FFT) où dans laquelle chaque acide aminé des séquences est représenté par un vecteur contenant les valeurs de volume et la polarité.

2) Une simplification du système de score pour avoir un temps de calcul réduit en faveur d'une recherche de l'exactitude soit pour les séquences de longues insertions et délétions soit pour des séquences divergentes de même longueur.

Deux heuristiques furent développées alors :

- Méthode progressive : (FFT-NS-2)
- Méthode itérative de raffinement (FFT-NS-i).

Le temps de la CPU a été sérieusement réduit par cette méthode en comparant avec les méthodes existantes.

II.3.2.2. Méthode T-Coffee

T-Coffee (Tree-based Consistency Objective Fonction for alignement Evaluation) [31] est une méthode qui essaye de pallier les problèmes de l'alignement progressif. Elle fait tout d'abord un prétraitement des données ; construction d'une bibliothèque [32] qui contient des alignements de paires de séquences fournis à partir de deux types d 'algorithmes d'alignement : global et local produits par deux méthodes connues (*ClustalW* et *Lalign de FASTA*).

En réalité T-Coffee réalise le même alignement progressif que ClustalW mais elle essaye d'échapper aux erreurs commises par ClustalW en utilisant des informations supplémentaires.

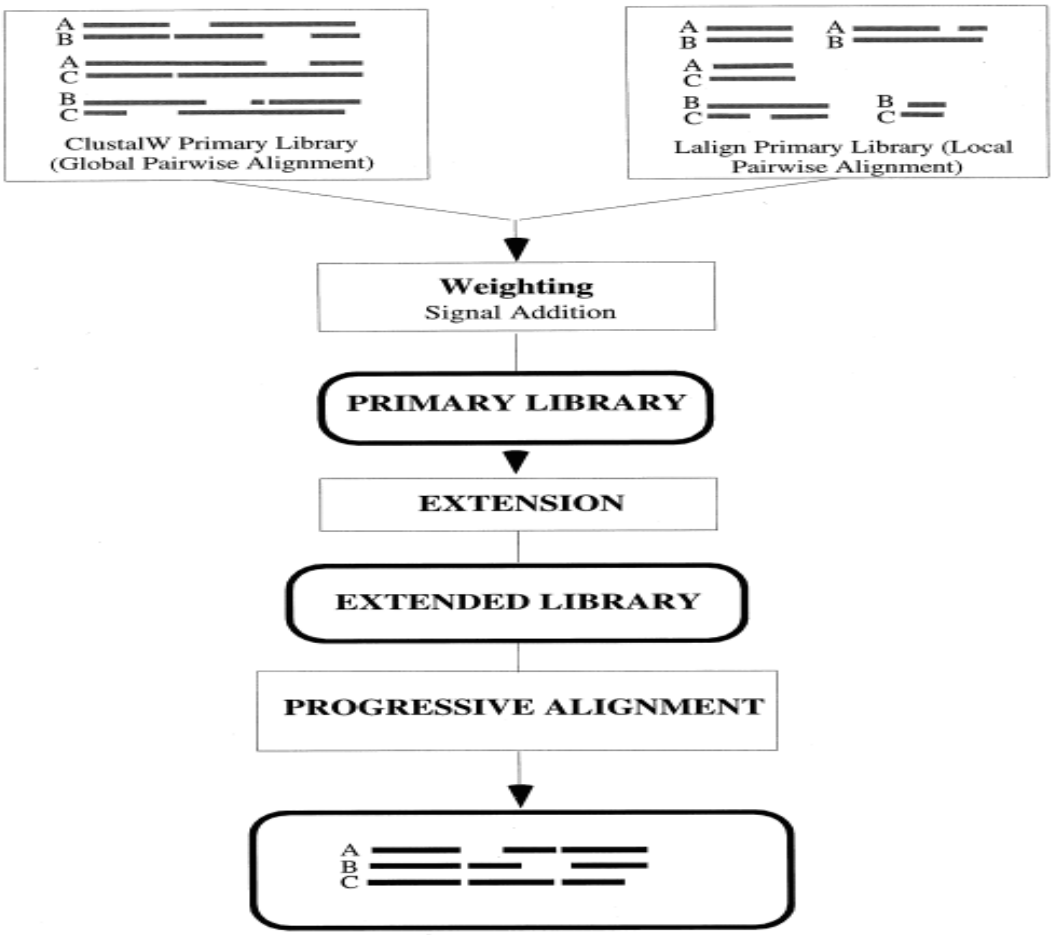

Figure 9 : le déroulement de T-Coffee [29]

- Produire des bibliothèques primaires des alignements :

§ Une bibliothèque concernant les alignements globaux produits par ClustalW

§ Une bibliothèque concernant les alignements locaux produits par Lalign.

§ Dans une bibliothèque, chaque alignement est représenté comme une liste de paires de résidus correspondants. Chaque paire de résidus dans la bibliothèque est considérée une contrainte à prendre en considération lors de l'évaluation de l'alignement.

- Déduire des poids de la bibliothèque :

Les poids dans chacune des bibliothèques sont calculés avec un pourcentage d'identité, une mesure qui est considérée être un indicateur raisonnable quand les séquences alignées ont plus que 30% d'identité.

- Combiner les bibliothèques ensemble dans la bibliothèque primaire :

Tous ces alignements contiennent de l'information qui est plus ou moins fiable. Par conséquent T-Coffee emploie leur combinaison pour confirmer la fiabilité des alignements. Le processus consiste alors à additionner les poids d'une paire de résidus si cette dernière apparaît dans les deux bibliothèques et ne garder qu'une seule entrée dans la bibliothèque finale.

- Extension la bibliothèque :

T-Coffee utilise une stratégie dont le but est de calculer les poids que reflète l'information contenue dans toute la bibliothèque. Pour le faire, on utilise *une approche* de triplet. Ceci fonctionne comme suit :

· Prendre chaque paire de résidus de la bibliothèque et de vérifier l'alignement de ces résidus avec les paires de résidus alignés dans les autres séquences.

· Soit la paire de résidus (Ai, Bj) avec A et B les séquences des résidus correspondants et i, j leur indices respectifs dans les séquences. Soit XAB le poids de l'alignement AB dans la bibliothèque primaire.

· Si Ai est aligné avec Ck et si Ck (C une troisième séquence) est aligné avec Bj avec des poids respectifs YAC et ZCB alors le poids de (Ai, Bj) sera calculée ainsi :

XAB ß XAB + Min (YAC, ZCB).

Ce nouveau poids sera temporairement le poids de Ai et Bj dans la bibliothèque étendue car il risque d'augmenter à chaque fois que l'on examine un nouveau triplet de séquences.

- Employez la bibliothèque étendue pour l'alignement progressif :

Afin de calculer l'alignement progressif nous calculons la matrice de distance en utilisant bibliothèque étendue. Elle est employée pour calculer un arbre guide en utilisant la méthode N.J.

Les gaps présents dans le premier alignement sont fixes et ne peuvent pas être décalées plus tard. En alignant deux groupes de séquences (contenant probablement seulement une séquence) *les scores* moyens de la bibliothèque étendue sont employés pour chaque colonne.

N.B. On remarque aussi que T-Coffee n'utilise pas une fonction objective proprement dite comme le fait la méthode SAGA.

La complexité est O(N3L2) avec N le nombre de séquences et L la longueur de l'alignement.

II.3.2.3. Méthode SAGA

C'est un algorithme génétique itératif [33] qui démarre par une population d'alignement, puis raffine les solutions par des opérateurs spécifiques tels que la mutation jusqu'à l'obtention d'une solution plus ou moins optimale. C'est une heuristique qui se rapproche de la solution optimale mais aucune certitude qu'elle le soit réellement.

Chaque génération est évaluée par la fonction objectif (*WSP*) pour déterminer quels sont les alignements les plus acceptables et aptes à passer dans la génération suivante. Ceci est appelé le phénomène de la sélection biologique

G0, Gn et Gn+1 sont respectivement la population initiale, courante et la population de la génération future. L'algorithme commence par la génération des individus de la population G0 d'une façon aléatoire, qui vont subir immédiatement une évaluation afin de déterminer le niveau de ces solutions. Si les solutions obtenues ont atteint un seuil d'optimalité alors l'algorithme s'arrête sinon on passe à l'étape suivante et qui consiste en la génération de nouvelles solutions en faisant subir à la population courante une série d'opérations génétiques telles que la sélection, croisement et mutation. Les nouvelles solutions obtenues ne sont maintenues dans la nouvelle génération que si elles présentent un certain niveau d'efficacité.
L'algorithme s'arrête après un certain nombre d'itération. La meilleure solution de la dernière population serait considérée la solution optimale de l'algorithme.

SAGA a la particularité de pouvoir optimiser n'importe quelle fonction objective. Plus tard

[30] ont utilisé SAGA pour valider une nouvelle fonction objective : Coffee.Les résultats sont considérés nettement meilleurs que ceux fournis par la première approche.

II.3.2.4. Méthode DIALIGN

DIALIGN est une méthode pour l'alignement multiple développée par [34]. L'algorithme de Dialign est basé sur les alignements par paires de séquence (alignement deux à deux) et multiple en comparant des segments entiers de séquences au lieu d'une traditionnelle comparaison de chaque résidu.
Des alignements par paires sont construits de pairs segments de même longueur sans insertion ou délétion de gaps. Ces paires de segments s'appellent les 'diagonales' ou (motif) observable sur le graphe d'un DOTPLOT. Par conséquent DIALIGN n'emploie aucune pénalité de gap.
Une fois une diagonale est considérée dans un alignement, elle est fixe et ne peut pas être enlevée à une étape postérieure de l'algorithme. Une diagonale n'est pas choisie selon son poids,

mais plutôt selon si le motif décrit par cette diagonale, apparaît dans plus de deux séquences, alors il est préféré aux motifs qui apparaissent dans seulement deux séquences.

Cette approche est particulièrement efficace et convenable pour la détection d'une homologie locale. Sa consommation en termes de durée de calcul et en espace mémoire est considérée raisonnable [35]. Dialign-t [36] est une version plus récente de Dialign-2, locale et progressive.

II.3.3. Méthodes Progressive

L'alignement progressif [37] est l'heuristique la plus répandue pour aligner un grand nombre de séquences. L'alignement multiple est construit progressivement en alignant des paires de séquences suivies des paires d'alignements/profils. Un arbre guide détermine l'ordre dans lequel les séquences vont être alignées, les plus proches d'abord. Cette technique est employée dans différents packages d'alignement multiple tels que MULTALIGN [38], ClustalW [39], et T-Coffee [40] …etc.

Un alignement multiple progressif suit les étapes suivantes [41] :

· Alignement deux à deux de toutes les séquences.

· Construction d'une matrice de distances entre toutes les séquences.

· Détermination de l'ordre selon lequel les séquences seront alignées en utilisant la notion de clustering :

- Alignement de deux séquences
- Alignement d'une séquence et d'un profil
- Alignement de deux profils

Problèmes majeurs des alignements multiples progressifs :

· Les alignements entre sous-groupes sont gelés. Si une erreur est produite au début, aucune modification ou correction ultérieure n'est possible.

Les erreurs dans les alignements des sous-groupes initiaux se propagent dans tous l'alignement.

II.3.3.1. Méthode ClustalW

ClustalW [42] est un programme qui met en action les principes de l'alignement progressif tout en essayant d'échapper au piège des erreurs qui peuvent se produire au début de l'alignement et nuire à sa qualité dans la fin.

Dans ClustalW, les auteurs essayent donc de respecter la démarche progressive mais en apportant des modifications et des nouvelles considérations.

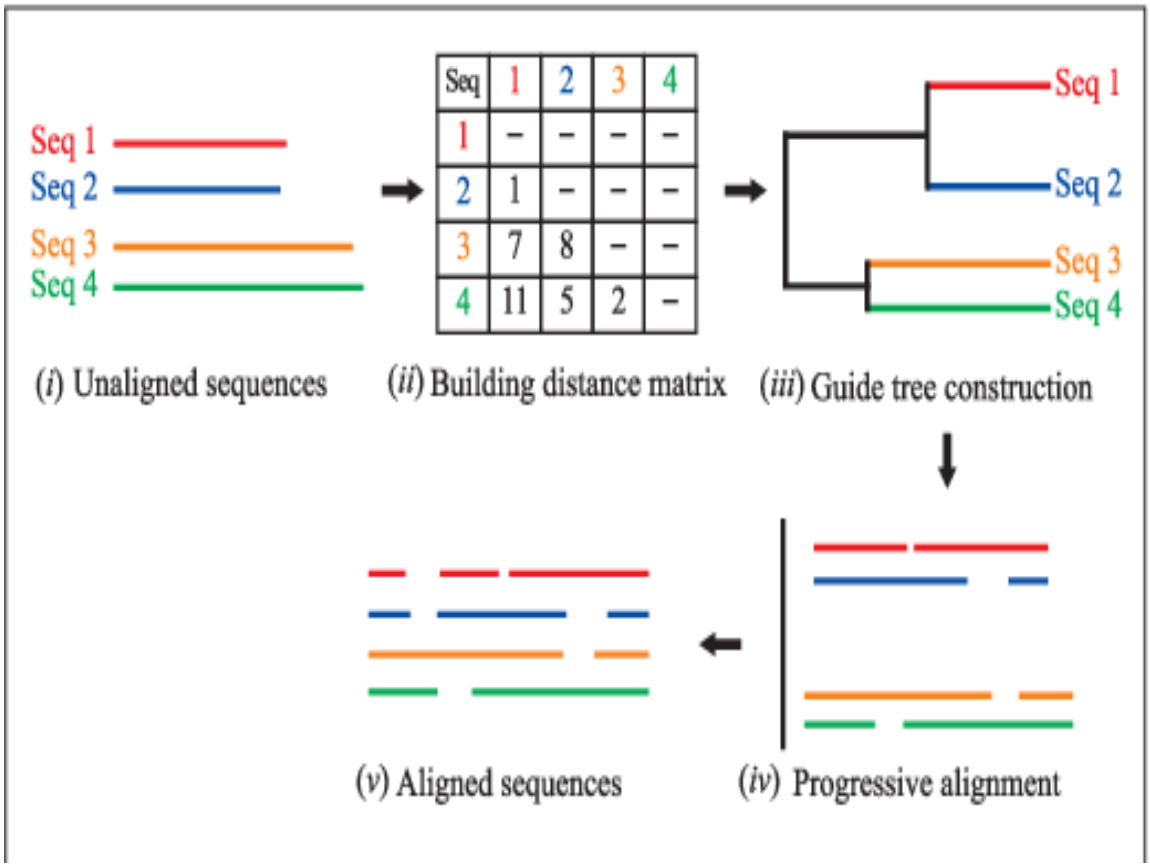

Figure 10 : Les étapes de la fonction Clustalw

La première étape de ClustalW (Fig.10) consiste à aligner les paires de séquences afin de déterminer la matrice des distances. ClustalW utilise des matrices de substitutions différentes pour la programmation dynamique à des moments différents de l'alignement. Les matrices changent selon la divergence ou la convergence des deux séquences à aligner. L'avantage est que les séquences divergentes sont plus ou moins bien alignées.

Dans la *deuxième étape*, ClustalW utilise la méthode N.J [43] pour construire un arbre guide et calculer les poids des séquences.

Pendant la *troisième étape* : alignement progressif proprement dit, ClustalW n'affecte pas la même valeur de pénalité d'un gap quel que soit sa position dans la séquence mais essayent de distinguer entre les gaps du début, du milieu et de la fin de la séquence.

Dans ClustalW, il y a une grande étude et des nouvelles propositions sur la manière de faire changer les valeurs affectées à un gap selon sa position dans une séquence ou dans un alignement de séquences.

Une particularité de ClustalW est qu'il possède une interface graphique conviviale contrairement aux autres méthodes

II.3.3.2. Méthode MUSCLE

La méthode MUSCLE [44] emploie deux mesures de distance pour une paire de séquences : une distance de k-mer de (pour une paire non alignée) et le Kimura distance (pour une paire alignée). Un k-mer est une subséquence contiguë de longueur k également connu sous le nom de mot ou k-tuplet. Les séquences homogènes possèdent plus de k-mers en commun que prévu par hasard. Cette mesure n'exige pas un alignement, elle donne un avantage significatif de vitesse contrairement à Kimura.

La méthode MUSCLE peut être décrite en trois étapes essentielles (Figure 10) :

L'étape 1 : Le but de la première étape est de produire rapidement un alignement multiple avec plus d'exactitude possible. Ceci est basé sur la détermination d'une matrice D1 de distances à partir de la distance de k-mers entre toutes les paires de séquences.

La matrice obtenue est alors clustérisée par UPGMA, pour produire un arbre binaire TREE1. Un alignement progressifMSA1 est construit alors en suivant l'ordre dicté par l'arbre.

L'étape 2 : La source d'erreur principale à l'étape progressive est la mesure approximative de distances k-mer, qui a comme conséquence un arbre sous optimal. MUSCLE re-estime donc l'arbre en utilisant la distance de Kimura, qui est plus précise mais exige l'utilisation un alignement dans ce cas c'est MSA1 donnant ma matrice D2. D2 va subir le même procédé de

Clustérisation afin de produire un arbre binaire TRRE2 et progressivement construire l'alignement MSA2

L'étape 3 : C'est une étape d'amélioration. TREE2 est divisé en deux sous arbres en supprimant la branche qui les relient Celle-ci est choisie en parcourant l'arbre à partir de la racine. Le profil de l'alignement multiple dans chaque sous arbre est alors calculé. Un nouvel alignement multiple a produit en réalignant les deux profils.

Si le score de PS est amélioré, le nouvel alignement est gardé, autrement il est rejeté et l'étape 3 est alors répétée jusqu' à la convergence ou jusqu'à ce qu'une limite définie soit atteinte.Considérée la plus rapide et plus exacte, la méthode MUSCLE est la plus répandue actuellement avec ClustalW.

II.3.3.3. ClustalOmega

Clustal Omega1 est un package pour réaliser des alignements de séquences multiples (MSA). Il a été développé il y a près de dix ans en réponse à l'augmentation considérable du nombre de séquences disponibles et à la nécessité d'effectuer de grands alignements rapidement et avec précision. Les paquets les plus utilisés pour faire des MSA au cours des 30 dernières années ont été Clustal W2 et Clustal X3, mais plus d'une centaine de paquets MSA ont été publiés pendant cette période. Ils se répartissent grosso modo en deux groupes principaux : ceux qui sont rapides et capables de réaliser de très grands alignements ou ceux qui sont plus précis et limités à un plus petit nombre de séquences. MUSCLE4 et MAFFT5 sont des exemples très utilisés du premier, tandis que T-Coffee6 et MAFFT L-INS-i7 sont des exemples du second. Clustal W et Clustal X sont largement utilisés en raison de leur large disponibilité sur les ordinateurs personnels et sur les serveurs et en raison de la robustesse et de la portabilité du code ainsi que de l'interface utilisateur très flexible et intuitive. Notre motivation initiale, lors de la conception de Clustal Omega, était de réaliser un ensemble capable de réaliser de très grands alignements sans sacrifier la précision.

Le second développement majeur de Clustal Omega a été l'utilisation d'un moteur d'alignement pour aligner les modèles de Markov cachés (HMM) entre eux au lieu de la programmation dynamique traditionnelle et de l'alignement de profil. Nous avons utilisé le HHalign11 qui s'est avéré très précis pour l'alignement du profil HMM. Cela donne beaucoup plus de précision à Clustal Omega par rapport aux programmes Clustal précédents, tels que mesurés sur des benchmarks d'alignement basés sur la structure. Seule une petite quantité de code original des programmes Clustal précédents a été utilisée pour le nouveau programme : les routines d'alignement par paires rapides basées sur des mots. Le reste du code a été codé à partir de zéro ou tiré de bibliothèques accessibles au public.

Cela a donné un tout nouveau programme capable d'aligner plusieurs milliers de séquences sans perdre de précision. Il a été publié en 2011 et est disponible gratuitement en téléchargement de tout le code source sous une licence Open Source. Les utilisateurs peuvent également télécharger des exécutables pour la plupart des systèmes d'exploitation (www. Clustal. Org) ou utiliser le logiciel en ligne sur de nombreux sites, notamment l'Institut Européen de Bioinformatique de l'EMBL (www. Ebi. Ac. uk).

II.3.4. Méthodes basées sur la consistance

II.3.4.1. Méthode PCMA

PCMA (Profile Consistency Multiple Sequence Alignment) [45] est programme progressif d'alignement multiple des séquences qui combine deux stratégies d'alignement. Des séquences fortement semblables sont alignées d'une manière rapide comme dans ClustalW, constituant les groupes pré-alignés. La méthode T-Coffee est appliquée pour aligner les groupes relativement divergents, elle est basée sur la comparaison et la consistance (consistency) profil-profil. La fonction de score pour les groupes pré-alignés est basé sur une nouvelle méthode de comparaison de profil-profil qui est une généralisation de l'approche de PSI-blast [46] de la comparaison profil- séquence. PCMA équilibre la rapidité et l'exactitude d'une manière flexible et convient à aligner un grand nombre de séquences.

PCMA est une méthode effectue en deux étapes :

La première étape : si deux séquences voisines quelconques ou groupes pré-alignés ont une moyenne d'identité par paire de séquences au-dessus d 'un certain seuil, par exemple 40%, elles sont alignées par l'algorithme de ClustalWpour constituer un nouveau groupe pré-aligné.

A la fin de la première étape, les séquences semblables forment des groupes pré-alignés avec une similitude relativement basse entre groupes voisins.

La deuxième étape : une mesure de consistance (Consistency) est appliquée (génération et extension de la bibliothèque) aux groupes pré-alignés, d'une manière semblable comme dans le programme de T-Coffee. Après la mesure de la consistance par l'extension de la bibliothèque, les groupes pré-alignés sont progressivement alignés les uns avec les autres en optimisant une fonction objective pour former l'alignement final.

La fonction de score utilisée pour évaluer les alignements locaux est basée sur une nouvelle méthode de comparaison profil-profil COMPASS (Comparison Of Multiple Protein Alignments With Assessment of Statistical Significance). Cette fonction construit des alignements profil–profil locaux optimaux et évalue analytiquement les E-values pour les similitudes détectées. Le système de score et le calcul de *E*-value sont basés sur une généralisation de l'approche de PSI-blast [46] pour la comparaison profil-séquence.

II.3.4.2. Méthode ProbCons

ProbCons [47] est une nouvelle méthode et un outil pratique pour l'alignement multiple progressif de séquences protéiques basé sur la probabilité de consistance (Consistency probability). ProbCons réalise statistiquement une amélioration significative par rapport à d'autres méthodes tout en préservant une vitesse pratique.

La probabilité de consistance est une nouvelle fonction de score pour des comparaisons des séquences multiples. ProbCons optimise la fonction basée Consistance (Consistency based) mais construit sur des modèles probabilistes selon les modèles cachés de Markov.

- **Le Modèle Caché de Markov (HMM : Hidden Markov Model)**

  Le modèle caché de Markov est un modèle stochastique composé de grand nombre d'états reliés entre eux, où chaque état émis un symbole observable. Les probabilités d'émission des symboles sont les probabilités d'émission possible de chaque symbole par un état. Cette séquence d'états est cachée et seulement la séquence de symbole émise est observable [48]. Les probabilités de transition d'état sont les probabilités de se déplacer de l'état actuel à un nouvel état en utilisant la distribution stochastique déterminée par l'état de la chaîne cachée de Markov. Dans le cadre de MSA, le modèle caché de Markov (HMM) fournit une formulation alternative du problème d'alignement de séquences dans lequel, la génération d'un alignement est directement modelée comme un processus de Markov de premier ordre impliquant des Émissions et des transitions d'états.

Le chemin le plus probable pour aligner une séquence sur un profil HMM est trouvé par L'algorithme de Viterbi. HMM peut simultanément trouver un alignement et un modèle de Probabilité des substitutions, insertions et délétions, qui est le plus consistant. Pour construire un l'alignement multiple, il faut calculer pour chacun séquence un alignement individuel de Viterbi [49].

II.3.5. Méthodes d'optimisation évolutionnaires et méthodes basées sur l'intelligence par essaim :

II.3.5.2. Optimisation par algorithmes génétiques :

Les algorithmes génétiques [50] (AG) sont des algorithmes d'optimisation stochastique fondés sur les mécanismes de la sélection naturelle et de la génétique. Ils ont été adaptés à l'optimisation par John Holland, également les travaux de David Goldberg ont largement contribué à les

enrichir Le vocabulaire utilisé est le même que celui de la théorie de l'évolution et de la génétique, on emploie le terme individu (solution potentielle), population ensemble de solutions), génotype (une représentation de la solution), gène (une partie du génotype), parent, enfant, Reproduction, croisement, mutation, génération, etc.

Leur fonctionnement est extrêmement simple, on part d'une population de solutions potentielles (chromosomes) initiales, arbitrairement choisies. On évalue leur performance (Fitness) relative. Sur la base de ces performances on crée une nouvelle population de solutions potentielles en utilisant des opérateurs évolutionnaires simples : la sélection, le croisement et la mutation. Quelques individus se reproduisent, d'autres disparaissent et seuls les individus les mieux adaptés sont supposés survivre. On recommence ce cycle jusqu'à ce qu'on trouve une solution satisfaisante. En effet, l'héritage génétique à travers les générations permet à la population d'être adaptée et donc répondre au critère d'optimisation, la figure 2.6 illustre les principales étapes d'un algorithme génétique.

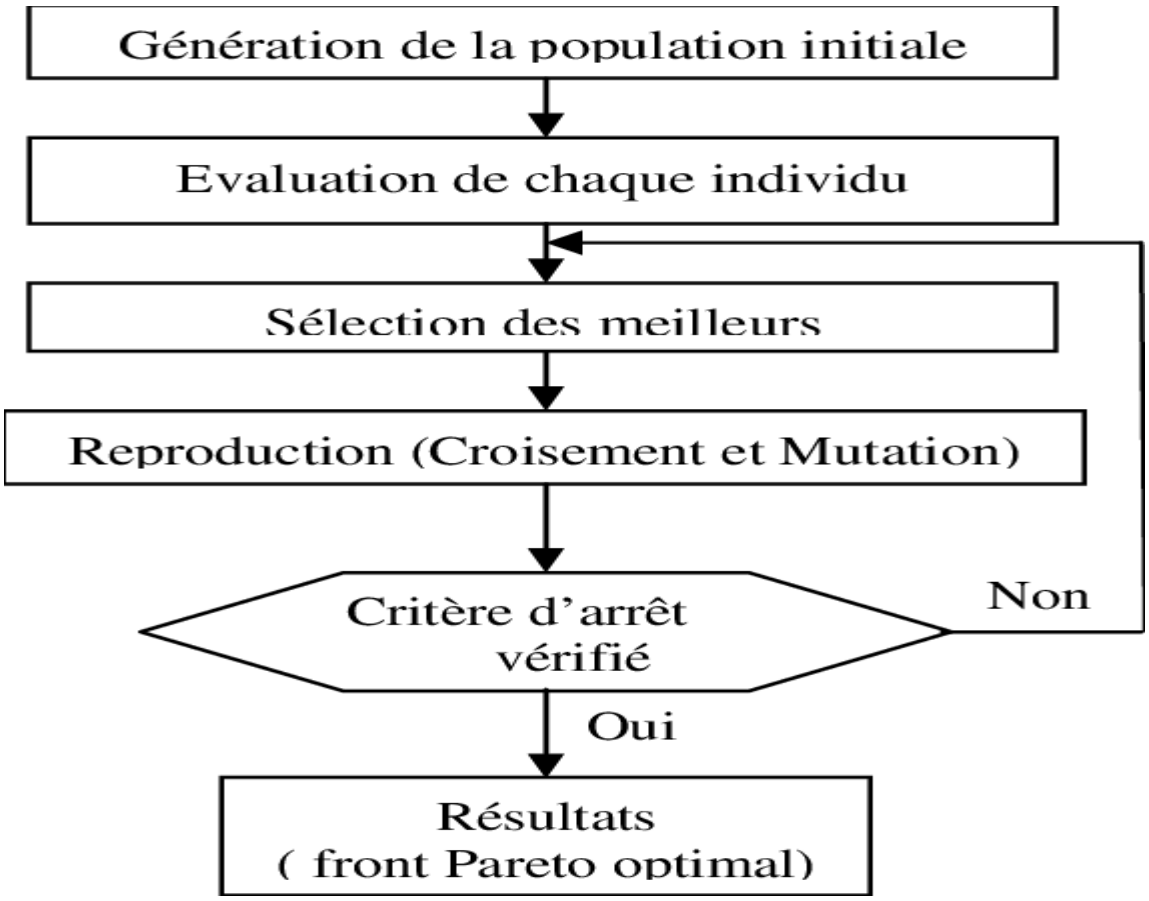

Figure 12 : L'algorithme génétique

II.3.5.1. Optimisation par essaim de particules (PSO)

L'optimisation par essaim de particules (le PSO en anglais : Particle Swarm Optimazation) est une métaheuristique à base de population de solution. Elle a été proposée en 1995 par Kennedy et Eberhart [48]. L'algorithme PSO est inspiré du comportement social d'animaux évoluant en essaim, tels que les poissons qui se déplacent en bancs ou les oiseaux migrateurs. En effet, on peut observer chez ces animaux des dynamiques de déplacement relativement complexes, alors qu'individuellement chaque individu a une intelligence limitée et une connaissance seulement locale de sa situation dans l'essaim. L'intelligence globale de l'essaim est donc la conséquence directe des interactions locales entre les différentes particules de l'essaim. La performance du système entier est supérieure de la somme des performances de ses parties. Kennedy et Eberhart se sont inspirés de ces comportements sociaux pour créer l'algorithme PSO. Contrairement aux autres algorithmes évolutionnaires tel que l'algorithme génétique où la recherche de la solution optimale évolue par compétition entre les individus en utilisant des opérateurs de croisements et de mutations, le PSO utilise plutôt la coopération entre les individus.

La méthode d'optimisation par essaim particulaire met en jeu un ensemble d'agents pour la résolution d'un problème donné. Cet ensemble est appelé essaim. L'essaim est composé d'un ensemble de membres, ces derniers sont appelés particules. Les particules de l'essaim représentent des solutions potentielles au problème traité. L'essaim de particules survole l'espace de recherche, en quête de l'optimum global. Le déplacement de chaque particule est influencé par les trois composantes suivantes [49]

- Une composante physique : la particule tend à suivre sa direction de déplacement courante ;
- Une composante cognitive : la particule tend à se diriger vers le meilleur site par lequel elle est déjà passée ;
- Une composante sociale : la particule tend à se diriger vers le meilleur site déjà atteint par ses voisines.

Chaque particule i de l'essaim est définie par sa position xid= (xi1, xi2, xid, xiD) et sa vitesse de déplacement vid= (vi1, vi2, vid, viD) dans un espace de recherche de dimension D.

Cette particule garde en mémoire la meilleure position par laquelle elle est déjà passée et la meilleure position atteinte par toutes les particules de l'essaim, notées respectivement :

PbestiD = (pbesti1, pbesti2, pbestid, pbestiD) et gbest = = (gbesti1, gbesti2, gbestid, gbestiD)

Le processus de recherche est basé sur deux règles :

- Chaque particule est dotée d'une mémoire qui lui permet de mémoriser la meilleure position par laquelle elle est déjà passée et elle a tendance à retourner vers cette position.

-Chaque particule est informée de la meilleure position connue au sein de son voisinage et elle a toujours tendance de se déplacer vers cette position.

La particule i va se déplacer entre les itérations t et t+1, en fonction de sa vitesse et des deux meilleures positions qu'elle connaît (la sienne et celle de l'essaim)

II.4. Conclusion

Dans ce chapitre nous avons présenté le problème des alignements multiples, en suite on a abordé les différentes méthodes de résolution de ces problèmes d ainsi que la description de chaque méthode. Dans ce qui suit, nous allons présenter la méthode proposée. Il s'agit d'une application de la métaheuristique de recherche coucou (CS pour résolution du problème d'alignement multiple. Nous décrivons les différentes étapes pour trouver le meilleur alignement et terminons par comparer les résultats obtenus par cette méthode avec d'autres.

Chapitre III : La Méthode proposée (CS-MAS)

III .1. Introduction

La résolution d'un problème d'optimisation consiste à rechercher une solution d'une qualité suffisante parmi un ensemble de solutions. Les méthodes de résolution de problèmes d'optimisation sont nombreuses. Nous avons appliqué une de ces méthodes qui est notre méthode proposé la recherche coucou sur le problème d'alignement multiple des séquences pour savoir si cette méthode est considérées comme un bon choix de résolution. Donc nous avons comparé les résultats de la recherche coucou avec d'autre méthode.

III .2. La recherche Coucou (CS)

En 2009, Xin-She Yang et Suash Deb ont proposé une nouvelle métaheuristique nommée la recherche coucou (CS). La recherche coucou est une métaheuristique très récente. Elle a enrichi le nombre des métaheuristiques à base de population de solutions. C'est une des variantes de l'algorithme d'optimisation par essaim particulaire (PSO). Les pionniers de l'algorithme CS se sont inspirés du comportement de reproduction parasitaire de quelques espèces de coucous qui pondent leurs œufs dans les nids des autres espèces en confiant la responsabilité d'incubation, de nourriture et d'élevage de leurs poussins aux oiseaux hôtes. Ces derniers peuvent détecter les œufs coucous dans leurs nids, dans ce cas-là l'oiseau hôte va ou bien éjecter l'œuf coucou or son nid ou abandonner son propre nid et construire un autre dans un autre emplacement. Yang et Deb se sont basés sur ce comportement parasitaire des coucous et sur le mécanisme du vol de Lévy qui permet la modélisation mathématique des déplacements aléatoires pour proposer une nouvelle méthode d'optimisation : La recherche coucou (CS). Malgré le nombre limité des travaux d'applications du CS [53] pour la résolution des problèmes d'optimisation, les résultats obtenus sont prometteurs en termes de performance et d'efficacité de la nouvelle métaheuristique dont le principe est élucidé dans ce qui suit.

III .3. Le principe et les étapes de la recherche coucou

En s'inspirant du comportement des coucous dans leur reproduction, Yang et Deb se sont basés sur trois principes pour proposer leur nouvelle métaheuristique [54].

- ✓ Chaque coucou pond un seul œuf à la fois. Il le dépose dans un nid qu'il choisit aléatoirement.

- ✓ Les meilleurs nids qui incluent des œufs (solutions) de bonnes qualités vont être les élus qui construisent les membres de la nouvelle génération.
- ✓ Le nombre des nids hôtes valides est fixé. L'oiseau hôte peut détecter le coucou étranger avec une probabilité $P_a \in [0,1]$. Dans ce cas-là, l'oiseau hôte tranche entre écarter le coucou de son nid en lui éjectant hors nid ou abandonner son nid pour aller construire un autre dans une nouvelle position.

La probabilité P_a représente la fraction de N nids qui vont être remplacés par de nouveaux nids (avec de nouvelles solutions aléatoires dans de nouvelles positions dans l'espace de recherche). La qualité d'un nid ou d'une solution est mesurée en fonction de la fonction fitness qui se varie d'un problème à un autre.

III .4. Utilisation de CS pour alignement multiple de séquences

Dans cette contribution, nous avons utilisé l'algorithme CS pour l'Alignement Multiple des Séquences (AMS). Pour cela nous avons adapté l'algorithme CS selon des propriétés du problème AMS, vu que AMS traite un ensemble de caractères.

L'algorithme 3.1 présente les étapes de la méthode proposée, nommée CS_MAS. Pour appliquer l'algorithme de coucou search nous avons passé par les étapes suivantes :

- ➢ **Trouver le meilleur coucou** : nous avons commencé par l'initialisation de la population puis nous avons calculé tous les coucous de la population pour trouver le meilleur coucou (g-best)
- ➢ **Déterminer le critère d'arrêt :** pour organiser le travail de notre algorithme nous avons mettre un critère d'arrêt comme suite :

Pour i de 1 a nombre max d'itération

- ➢ **Générer un poussin :** pour créer le poussin nous avons utilisé la fonction RANDOM pour modifier l'ancien coucou par le remplacement des positions des gaps donc nous avons obtenus une nouvelle instance. Cette instance nous donne une nouvelle population donc un nouveau score. Ce score on va le comparer avec l'ancien coucou.

- **Remplacement le coucou par son poussin :** Apres l'évaluation des deux solutions (poussin et coucou) nous avons trouvé que le poussin est meilleur que le coucou donc :

Si le poussin est meilleur que le coucou
Remplacer le coucou par le poussin

- **Trouver G-BEST :** Dans notre programme nous avons inclus une boucle. Cette boucle va répéter à chaque fois l'évaluations des solutions et le remplacement par le meilleur jusqu'à le nombre d'itérations déterminé (nombre d'itération max est 99).

A la fin le programme va afficher le GBEST final qui représente la meilleure solution dans toutes les itérations et il va afficher encore la ligne et l'instance de GBEST final.

Début

Créer une population de N coucous (solutions) ;
Présenter les coucous créés par des décimales.

Tant que (le critère d'arrêt n'est pas satisfait) **faire**

Pour chaque coucou s **faire**
Créer son poussin g ;
Calculer le score (la fitness) de s et de g ;
Remplacer s par g si f (g) est meilleur que f (s) ;
Fin pour

Trouver le meilleur coucou ;

Pour chaque coucou s **faire**

Modifier une fraction P_a de son contenu pour obtenir une nouvelle solution s' ;
Calculer le score de s';
Remplacer s par s'si f (s') est meilleur que f (s) ;

Fin pour

Trouver le meilleur coucou ;

Fin tant que

Décoder la meilleure solution

Retourner le meilleur alignement trouvé ;

Fin

Algorithme 3.1. Algorithme de recherche coucou pour AMS

III .4.1. Création de la population

Nous avons utilisé la méthode CLUSTALW pour créer une population de bonne qualité.la méthode ClustalW permet de proposer des alignements possibles pour un ensemble de séquences en insérant des gaps tout au long des séquence à aligner comme le montre les figures 1 et 2.

```
>1aboA

NLFVALYDFVASGDNTLSITKGEKLRVLGYNHNGEWCEAQTKNGQGWVPSNYITPVN
>1ark
TAGKIFRAMYDYMAADADEVSFKDGDAIINVQAIDEGWMYGTVQRTGRTGMLPANYVEAI
>1gbq
MEAIAKYDFKATADDELSFKRGDILKVLNEECDQNWYKAELNGKDGFIPKNYIEMKP
>1ckb
AEYVRALFDFNGNDEEDLPFKKGDILRIRDKPEEQWWNAEDSEGKRGMIPVPYVEKY
>1gfc
GSTYVQALFDFDPQEDGELGFRRGDFIHVMDNSDPNWWKGACHGQTGMFPRNYVTPV
>1hsp
GSPTFKCAVKALFDYKAQREDELTFIKSAIIQNVEKQEGGWWRGDYGGKKQLWFPSNYVE
EMV
>1aey
GKELVLALYDYQEKSPREVTMKKGDILTLLNSTNKDWWKVEVNDRQGFVPAAYVKKL
>1csk
GTECIAKYNFHGTAEQDLPFCKGDVLTIVAVTKDPNWYKAKNKVGREGIIPANYVQKR
>1ad5
EDIIVVALYDYEAIHHEDLSFQKGDQMVVLEESGEWWKARSLATRKEGYIPSNYVARVD
>1awj
RRSFQEPEETLVIALYDYQTNDPQELALRCDEEYYLLDSSEIHWWRVQDKNGHEGYAPSS
YLVEKS
>1efn
ALFVALYDYEAITEDDLSFHKGEKFQILNSSEGDWWEARSLTTGETGYIPSNYVAPV
>1sem
ETKFVQALFDFNPQESGELAFKRGDVITLINKDDPNWWEGQLNNRRGIFPSNYVCPY
>1ycsB
KGVIYALWDYEPQNDDELPMKEGDCMTIIHREDEDEIEWWWARLNDKEGYVPRNLLGLYP
>1pht
GYQYRALYDYKKEREEDIDLHLGDILTVNKGSLVALGFSDGQEARPEEIGWLNGYNETTG
ERGDFPGTYVEYIGRKKISP
>1ihvA
    NFRVYYRDSRDPVWKGPAKLLWKGEGAVVIQDNSDIKVVPRRKAKIIRD
```

Figure 16 : instance avant alignement

```
1ad5        -------EDIIVVALYDYEAIHHEDLSFQKGDQMVVL--------------EES--GEWW
1efn        ---------ALFVALYDYEAITEDDLSFHKGEKFQIL--------------NSSE-GDWW
1aboA        ---------NLFVALYDFVASGDNTLSITKGEKLRVL--------------GYNHNGEWC
1gbq        ---------MEAIAKYDFKATADDELSFKRGDILKVL--------------NEECDQNWY
1csk        --------GTECIAKYNFHGTAEQDLPFCKGDVLTIV--------------AVTKDPNWY
1gfc        -------GSTYVQALFDFDPQEDGELGFRRGDFIHVM--------------DNSD-PNWW
1sem        -------ETKFVQALFDFNPQESGELAFKRGDVITLI--------------NKDD-PNWW
1ckb        --------AEYVRALFDFNGNDEEDLPFKKGDILRIR--------------DKPE-EQWW
1hsp        ---GSPTFKCAVKALFDYKAQREDELTFIKSAIIQNV--------------EKQE-GGWW
1ycsB        --------KGVIYALWDYEPQNDDELPMKEGDCMTII--------------HREDEDEIE
1aey        -------GKELVLALYDYQEKSPREVTMKKGDILTLL--------------NSTN-KDWW
1awj        RRSFQEPEETLVIALYDYQTNDPQELALRCDEEYYLL--------------DSSE-IHWW
1ark        ------TAGKIFRAMYDYMAADADEVSFKDGDAIINV--------------QAID-EGWM
1pht        --------GYQYRALYDYKKEREEDIDLHLGDILTVNKGSLVALGFSDGQEARPEEIGWL
1ihvA        ------NFRVYYRDSRDPVWKGPAKLLWKGEGAVVIQ----------------------
                        :       :

1ad5        KARSLATRKEGYIPSNYVARVD------
1efn        EARSLTTGETGYIPSNYVAPV-------
1aboA        EAQTKNG--QGWVPSNYITPVN------
1gbq        KA-ELN-GKDGFIPKNYIEMKP------
1csk        KA-KNKVGREGIIPANYVQKR-------
1gfc        KG-ACHG-QTGMFPRNYVTPV-------
1sem        EG-QLNN-RRGIFPSNYVCPY-------
1ckb        NA-EDSEGKRGMIPVPYVEKY-------
1hsp        RG-DYGGKKQLWFPSNYVEEMV------
1ycsB        WWWARLNDKEGYVPRNLLGLYP------
1aey        KV--EVNDRQGFVPAAYVKKL-------
1awj        RVQDKNG-HEGYAPSSYLVEKS------
1ark        YGTVQRTGRTGMLPANYVEAI-------
1pht        NGYNETTGERGDFPGTYVEYIGRKKISP
    1ihvA          -----DNSDIKVVPRRKAKIIRD-----
```

Figure 17 Instance après alignement avec clustalw

- **Représentation des coucous**

  Dans cette étape on a fait un codage binaire sur notre instance .Et pour arriver à ce résultat ;

  On a créer un code qui transforme les GAP en numero 1 et les caractéres en numero 0.

```
Début

    Pour chaque caractère
        Remplacer caractères par 0
    Pour chaque GAP
        Remplacer GAP par 1
    Fin pour
Fin
```

Figure 18 : Le pseudo code de transformation en binaire

Après l'exécution de ce code sur notre instance, le résultat de ce codage venir comme suivant :

```
-------EDIIVVALYDYEAIHHEDLSFQKGDQMVVL--------------EES--GEWW KARSLATRKEGYIPSNYVARVD------
---------ALFVALYDYEAITEDDLSFHKGEKFQIL--------------NSSE-GDWW EARSLTTGETGYIPSNYVAPV-------
---------NLFVALYDFVASGDNTLSITKGEKLRVL--------------GYNHNGEWC EAQTKNG--QGWVPSNYITPVN------
---------MEAIAKYDFKATADDELSFKRGDILKVL--------------NEECDQNWY KA-ELN-GKDGFIPKNYIEMKP------
--------GTECIAKYNFHGTAEQDLPFCKGDVLTIV--------------AVTKDPNWY KA-KNKVGREGIIPANYVQKR-------
-------GSTYVQALFDFDPQEDGELGFRRGDFIHVM--------------DNSD-PNWW KG-ACHG-QTGMFPRNYVTPV-------
```

Figure 19 : Exemple d'une partie de notre instance

```
11111110000000000000000000000000000011111111111111000110000000000000000000000000111111
11111111100000000000000000000000000011111111111110000100000000000000000000000011111111
11111111100000000000000000000000000011111111111100000000000000000110000000000000011111111
11111111100000000000000000000000000011111111111100000000000010001000000000000000001111111
11111111100000000000000000000000000011111111111110000000000010000000000000000000011111111
11111110000000000000000000000000000011111111111110000100000010000100000000000000011111111
```

Figure 20 : Le résultat d'exécution de code de transformation en binaire.

- **Le codage en forme décimale**

Nous avons construit un code qui transforme notre résultat précédent de codage binaire (figure 3.5) en forme décimal, et pour créer ce code nous avons traduisez chaque colonne de la matrice binaire en numéro décimale le résultat de ce codage venir comme suivant :

28	67	89	53	97	52	87	44	76	80

Figure 21 : Exemple de résultat de codage décimal.

La (figure 22) représente le code de transformation en décimale :

```
def binary_to_decimal(binary):
     binary = "«. join(reversed(binary))

decimal = 0
For number in range(len(binary)):
      If binary[number] == "1 »:
          decimal += pow (2, number)
Return decimal
```

Figure 22 : le code de transformation en décimale

- **Fonction de décodage en caractères**

Avant de commencer à calculer le sore (fitness) nous avons utilisé la fonction de décodage qui permet de transformer les coucous obtenus qui sont sous forme décimale vers la forme initiale (caractères et gaps)

Et la figure 23 représente cette fonction :

```
Def to Binary(a) :
         L, m= [], []
         For i in a :
               l.append(ord(i))
        Print(l)
        For i in l :
             m.append(int(bin(i) [2 :]))
Return m
```

Figure 23 : Le décodage en caractères

- **La fonction fitness**

Afin de mesurer la qualité d'un coucou, nous devons calculer le score de l'alignement qu'il propose. Pour cela nous avons utilisé une fonction pour mesurer le fitness des coucous. Les équations de (figure 24) ont été utilisées pour calculer le score des alignement multiples trouvés.

$$S = \sum_{l=1}^{L} S_l \ ;$$

$$\text{Avec} \quad S_l = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} w_{ij} cout(A_i A_j)$$

$$W_{ij} = Nombre\ de\ caractères\ différents\ dans\ l'alignement \Big/ Longueur\ totale\ de\ l'alignement$$

Figure 24 : les équations pour calculer le score des alignement multiples

S est le coût des alignements de séquences multiples. L est le nombre de colonnes dans tout l'alignement. S_l est le coût de la $l^{ème}$ colonne de longueur L. N est le nombre de séquences. W_{ij} est le poids des séquences i et j. Il définit la diversité entre deux séquences.

Cout (Ai, Aj) est le score d'alignement entre deux séquences alignées Ai et Aj.

- Ai ≠ « _ » et Aj ≠ « _ » alors le coût (Ai, Aj) est déterminé à partir du pourcentage de matrice de mutations acceptables.
- Ai = "_" et Aj = "_" alors coût (Ai, Aj) = 0.
- Ai = « _ » et Aj ≠ « _ » ou Ai ≠ « _ » et Aj = « _ » alors coût (Ai, Aj) = 1.

Enfin, la fonction de coût « coût (Ai, Aj) » comprend la somme des coûts de substitution de l'insertion ou des suppressions.

III .5. Dataset utilisés

Dans notre projet nous avons utilisé la référence 2 et la référence 3 de BAliBASE [55] qui se compose de 142 alignements de référence, contenant plus de 1000 séquences avec 200.000 résidus. Les alignements sont divisés en cinq catégories hiérarchiques de référence. Chacune des catégories peut être encore subdivisée en plus petits groupes, selon la longueur de séquence et les pourcentages de similitude :

La référence 1 : contient des alignements de (moins de 6) séquences équidistantes, dont le pourcentage d'identité entre deux séquences est dans une marge indiquée. Toutes les séquences sont de longueur semblable, pas de grandes insertions ou prolongements de gaps.

La référence 2 : aligne jusqu'à trois séquences orphelines (moins de 25% d'identité) de la référence 1 avec une famille d'au moins de 15 séquences très proches.

La référence 3 : se compose de familles de séquences équidistantes divergentes (un pourcentage d'identité <25%).

La référence 4 : est divisée en deux sous-catégories contenant des alignements de jusqu'à 20 séquences comprenant des prolongements de N/C-terminal (jusqu'à 400 résidus),

La référence 5 : contient des séquences de longues insertions internes (jusqu'à 100 résidus).

Dans BAliBASE des blocs ≪ *noyau* ≫ (corês) sont annotés pour les alignements qui incluent des régions qui doivent être correctement alignées. Les blocs excluent des régions où il y a une possibilité d'ambiguïté. Ceci peut être un facteur important affectant la signification des comparaisons statistiques des programmes d'alignement.

Référence 2	Référence 3
1aboA	1idy
1idy	1r69
1csy	1ubi
1r69	1uky
1tvxA	
1tgxA	
1ubi	
1wit	
2trx	
1sbp	
1havA	

Figure 25 : Les instances utilisées

III .6. Environnement et matériel utilisé

Dans toutes les étapes de notre travail nous avons utilisé un PC Asus SonicMaster avec les caractéristiques suivante :

- Possesseur RAYZEN 5
- 8 GO de RAM

- 256 GO SSD de stockage
- Carte graphique RADEON VEGA 8

Le système d'exploitation utilisée pour la réalisation de notre différente fonction est le système Windows 10 Professionnel c'est un système connu pour son efficacité, sa fiabilité, sa robustesse, et sa sécurité ainsi que la richesse de ses outils de programmations. Le langage de programmation choisi : python 3.9 (jupyter Notebook anaconda 3).

III.7. Résultats et comparaison

Pour tester la performance de la méthode proposée (dite PSO-MSA), nous l'avons appliquée sur 16 instances de la base de données BRALIBASE.2. En outre, nous avons comparé les résultats obtenus avec ceux d'autres méthodes proposées dans la littérature comme : HMMT, ML-PIMA, DIALI et PILEUP-8 [56].

Tableau 2 : Résultats obtenus avec des instances de Ref2.

Instance	**HMMT**	**ML-PIMA**	**DIALI**	**PILEUP-8**	**CS-MAS**
1aboA	0.724	0.220	0.384	0.000	**0.753**
1idy	0.353	0.000	0.000	0.000	**0.535**
1csy	0.000	0.000	0.000	0.114	**0.780**
1r69	0.000	**0.675**	**0.675**	0.450	**0.255**
1tvxA	0.276	0.241	0.000	0.345	**0.542**
1tgxA	0.622	0.543	0.630	0.318	**0.785**
1ubi	0.053	0.129	0.000	0.000	**0.738**
1wit	0.641	0.463	0.724	0.476	**0.756**
2trx	0.739	0.702	0.734	0.87	**0.762**
1sbp	0.214	0.054	0.043	0.177	**0.782**
1havA	0.194	0.238	0.000	0.493	**0.620**
1uky	0.395	0.306	0.216	0.562	**0.718**

DISSCUSION :

Dans ce tableau nous avons comparé notre résultat obtenu par (CS-MAS) avec d'autre méthodes telle-que (HMMT, ML-PIMA, DIALI, PILEUP-8) (avec des instances de REF2) on observe que le meilleur résultat (Moy-score) c'est le résultat de la méthode de coucou search (CS-MAS) dans toutes les instances avec un Moy-score totale de (0,668) sauf dans l'instance 4 (1r69) le

meilleur résultat obtenu c'était par deux méthode (ML-PIMA et DILI) avec la même valeur (0,675).

Tableau 3 : Résultats obtenu avec des instances de ref3.

Instance	HMMT	ML-PIMA	DIALI	PILEUP-8	CS-MAS
1idy	0.227	0.000	0.000	0.000	**0.489**
1r69	0.000	**0.905**	0.524	0.000	0.650
1ubi	0.366	0.000	0.000	0.268	**0.570**
1uky	0.037	0.148	0.139	0.083	**0.432**

Ce tableau montre les résultats obtenus avec des instances de ref3 on observe que le meilleur moy-score c'était par la méthode (CS-MAS) avec un score (0,535) même observation dans la comparaison avec toutes les instances sauf dans l'instances 1r69 c'était par la méthode ML-PIMA (0,905). Donc les résultats obtenus dans cette comparaison montrent que la méthode choisie est la plus performante parmi toutes les autres méthodes et montre aussi sa capacité à améliorer la qualité des alignements

Les résultats des tableaux sont présentés dans les graphes suivants :

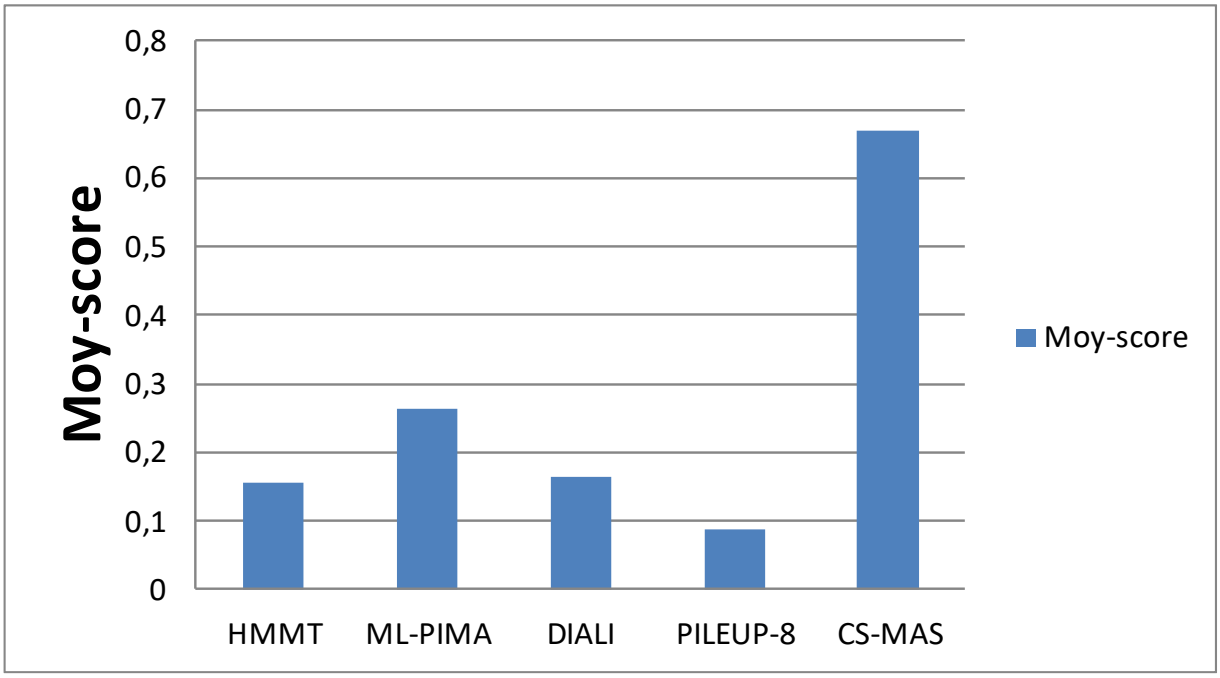

Figure 26 : performance des méthodes sur les instances

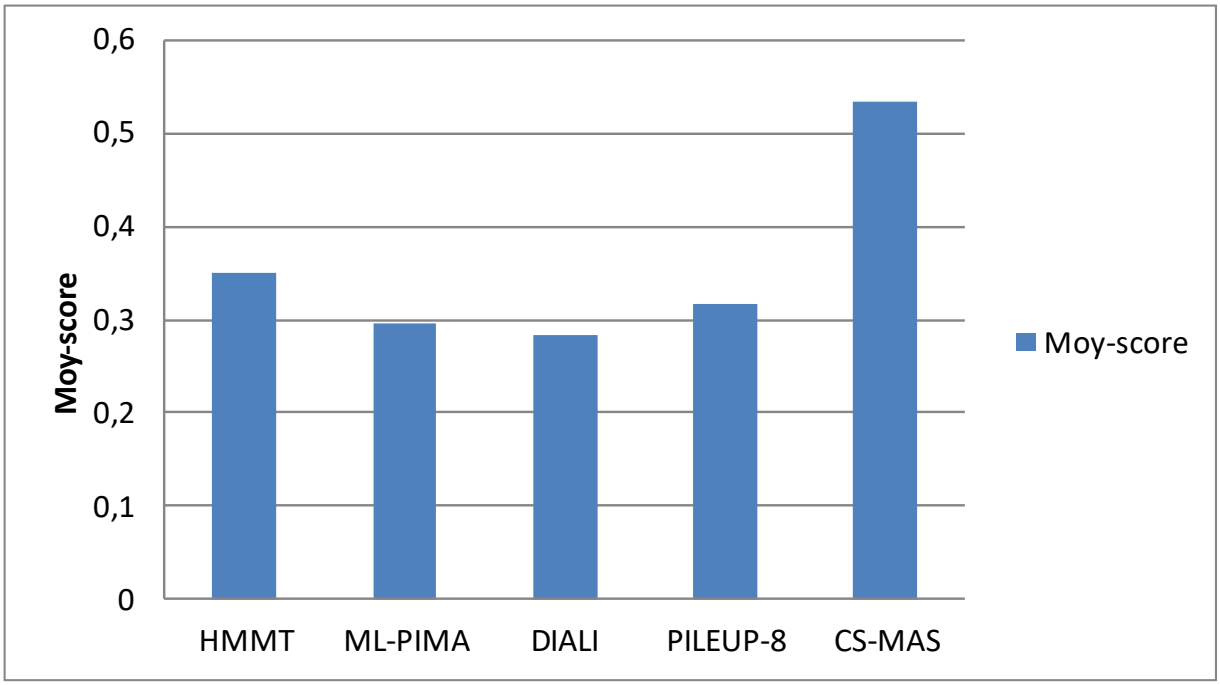

Figure 27 : performance des méthodes sur les instances

III .8. Conclusion

Dans ce chapitre, et en premier temps nous avons présenté la méthode de recherche coucou (CS) le principe de cette métaheuristique et ces différentes étapes. Ensuite, nous avons montré son adaptation pour l'alignement multiple de séquences. Afin d'évaluer notre méthode nous avons détaillé les taches réalisées dans chaque étape, décrit l'environnement et le matériel utilisé et cité les données utilisés (Dataset). Enfin nous avons présenté les résultats obtenus et les comparer avec des résultats d'autres méthodes.

Conclusion Général

Conclusion générale

La bioinformatique est un champ de recherche multidisciplinaire où travaillent des biologistes, informaticiens, mathématiciens et physiciens, dans le but de résoudre un problème scientifique ; Également, la bioinformatique joue un rôle important pour modéliser, analyser, comparer et simuler l'information biologique.

L'alignement de séquences multiples ou MSA (pour Multiple Sequence Alignment) est un problème important en biologie moléculaire et représente une tâche fondamentale pour de nombreuses applications en bioinformatique.

Par conséquent, plusieurs d'autres méthodes existent pour la résolution du problème d'MSA. Ces méthodes sont des méthodes approchées qui permettent de proposer des bons alignements avec des coûts de réponse raisonnables

Notre contribution été la proposition d'un algorithme basé CS pour l'alignement multiple des séquences. Nous avons proposé une adaptation des procédures de recherche et d'optimisation de l'algorithme CS aux caractéristiques du problème d'alignement multiple des séquences. Par ailleurs, nous avons implémenté un ensemble de fonctions en utilisant le langage de programmation Python. Ces fonctions travaillent en collaboration afin de réaliser un ensemble d'étapes permettant de proposer des solutions initiales et les optimiser au cours d'un ensembles d'itérations afin d'en sortir la meilleure solution qui représente le meilleur alignement multiple des séquences nucléiques ou protéiques. La performance de la méthode proposée a été mesuré par son application sur un ensembles d'instances d'une base de données publique nommée BRALIBASE qui permet d'accéder à un ensemble de séquences regroupées dans des fichiers. Les résultats obtenus par notre méthode ont été comparés par celles des autres méthodes connues dans la littérature. La comparaison a montré l'efficacité de la méthode proposée.

Références Bibliographiques

Références bibliographiques

[1] Thompson et autres, 94 J.D. Thompson, D.G. Higgins and T.J. Gibson, "CLUSTAL W : Improving the sensitivity of progressive multiple sequence alignment Through sequence weighting, position-specific gap penalties and weight matrix Choice", *Nucleic Acids Res*. Vol. 22 No. 22 pp. 4673-4680, 1994

[2] K. Katoh, K. Misawa, K. Kuma and T. Miyata, "MAFFT: a novel method for rapid Multiple sequence alignment based on fast Fourier transform", *Nucleic Acids Res*. Vol. 30 no. 14 pp. 3059-3066, 2002.

[3] J. Pei, R. Sadreyev and N.V. Grishin, "PCMA: fast and accurate multiple sequence Based profile consistency", *Bioinformatics.* Vol. 19, pp. 427-428, 2003.

[4] Kennedy et Eberhart, 1995. J. Kennedy, R.C. Eberhart. A discrete binary version of the particle swarm algorithm. Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics, NJ: Piscatawary. pp. 4104-4109, 1997

[5] Schneider V, Goujon C, Delphin N, Dutreuil C, Jacomet C. Intérêt du Séquençage dans le diagnostic virologique : application au VIH et au VHC. RevueFrançaise Des Laboratoires 1999 :39–44.

[6] Sanger F, Nicklen S, Coulson AR. DNA sequencing with chainterminating Inhibitors. Proc Natl Acad Sci USA 1977 : 5463-7.

[7] Lander ES, Linton LM, Birren B, Nusbaum C, Zody MC, Baldwin J, et al. Initial sequencing and analysis of the human genome. Nature 2001 :860—921.

[8] Venter JC, Adams MD, Myers EW, Li PW, Mural RJ, Sutton GG, et al. The Sequence of the human genome. Science 2001 :1304—51.

[9] P.Y. Chou and G.D. Fasman. Prediction of protein conformation. Biochemistery, 13 :222 245, 1974.

[10] M.O. Dayhoff, R.M. Schwartz et B.C. Orcutt. A model for evolutionary change in proteins. Atlas of Protein Sequence and Structure, 5 :345–352, 1978.

[11] J. Felsenstein. Inferring phylogenies. Sinauer Associates, 2004.

[12] Brown V, Jin P, Ceman S, et al. Microarray identification of FMRP-associated brain mRNAs and altered mRNA translational profiles in fragile X syndrome. Cell 2001; 107: 477–87.

[13]: M. Wallace, O. O'Sullivan, D. G. Higgins and C. Notredame. « M-Coffee: combining multiple sequence alignment methods with TCoffee", *Nucleic Acids Res.*, 2006, Vol. 34, No. 6, pp.1692–1699.

[14]: R.C Edgar, "MUSCLE: multiple sequence alignment with high accurac and high through put", *Nucleic Acids Res*. Vol. 32, No. 5, pp. 1792-1797, 2004.

[15]: Rong etHansen, Z. Rong and E.A. Hansen. "K-Group A* for Multiple Sequence Alignment with Quasi-Natural Gap Costs" 16th IEEE International Conference on Tools with Artificial Intelligence.

[16]: Wallace, 04 I. M. Wallace, O. O'Sullivan, D. G. Higgins and C. Notredame. «M-Coffee: combining multiple sequence alignment methods with T-Coffee", *Nucleic Acids Res.*, 2006, Vol. 34, No. 6pp.1692–1699.

[17]: Smith et Waterman, 81 T. Smith and M. Waterman, “Identification of common molecular subsequence”. *J. Mol. Biol.* Vol. 147, pp. 195-197. 1981.

[18]: J. Stoye, V, Moulton, and A. W. Dress, « DCA, an efficient Implementation of the divide and conquer approach to simultaneous multiple sequence Alignment”, Comput. Appl. Biosc., Vol. 13, No. 6, pp. 625-631, 1997.

[19]: A.H. Land, A.G. Doig automatic method for solving discrete programming problems Econometrica (1960), pp. 497-520

[20]: E.L. LAWLER, D. E Wood Branch-and-bound methods: A survey Oper. Res., 14 (1966), pp. 699-719

|22]: G. Nemhauser, L. Wolsey Integer and Combinatorial Optimization. Vol. 18 Wiley, New York (1988)

[23] :D. Bertsimas, J.N. Tsitsiklis Introduction to Linear Optimization Athena Scientific (1997)

[24]: C.H. Papadimitriou, K. Steiglitz Combinatorial Optimization: Algorithms and Complexity Courier Dover Publications (1998)

[25]: J. Clausen, Department of Computer Science, University of Copenhagen (1999) Branch and bound algorithms-princip les and examples. Technical Report

[26]: D.R. Morrison, E.C. Sewell, S.H. Jacobson an application of the branch, bound, and remember algorithm to a new simple assembly line balancing dataset European J. Oper. Res. (2013)

[27]: M. Guzelsoy, G. Nemhauser, M. Savelsbergh Restrict-and-relax search for 0–1 mixed-integer programs EURO J. Comput. Optim., 1 (2013), pp. 201-218.

[28]: Dechter, J. Pearl Generalized best-first search strategies and the optimality of A∗ J. ACM, 32 (1985),pp. 505-536

[29] : Layeb, 05 A. Layeb, « Approche quantique évolutionnaire pour l’alignement multiple de séquences en bioinformatique », mémoire de Magistère, Département d’Informatique, Université Mentouri Constantine,2005. [30]:

Stoye et autres, 97 J. Stoye, V, Moulton, and A. W. Dress, « DCA, an efficient implementation of the divide and conquer approach to simultaneous multiple sequence Alignment”, Comput. Appl. Biosc., Vol. 13, No. 6, pp. 625-631, 1997.

[31]: K. Katoh, K. Misawa, K. Kuma and T. Miyata, “MAFFT: a novel method for rapid Multiple sequence alignment based on fast Fourier transform”, *Nucleic Acids Res*. Vol. 30 no. 14 pp. 3059-3066, 2002.

[32]: Notredame et autres, 00 C. Notredame, L. Holm and D.G. Higgins, « Coffee: an objective function for multiple sequence alignments », *Bioinformatics*, Vol. 14, No. 5 pp. 407-422, 1998

[33] : Notredame et autres,98 C. Notredame and D.G. Higgins,” SAGA : Sequence Alignment by genetic algorithm”, *Nucleic Acids Res*. Vol. 24, No. 8 pp. 1515-1524, 1996.

[34]: Notredame et Higgins, 96 C. Notredame and D.G. Higgins,” SAGA: Sequence Alignment by genetic algorithm”, *Nucleic Acids Res*. Vol. 24, No. 8 pp. 1515-1524, 1996.

[35] : Notredame et autres,98 C. Notredame and D.G. Higgins,” SAGA : Sequence Alignment by genetic algorithm”, *Nucleic Acids Res*. Vol. 24, No. 8 pp. 1515-1524, 1996.

[36]: Morgenstern et autres, v98 B. Morgenstern, K. Frech, A. Dress and T. Werner, “DIALIGN: Finding local similarities by multiple sequence alignment”, *Bioinformatics*, Vol. 14, No. 3 pp. 290-294, 1998.

[37]: Lambert, 03 C. Lambert, J. V. Campenhout, X. DeBolle and E. Depiereux, “Review of Common Sequence Alignment Methods: Clues to Enhance Reliability”, *Current Genomics*, vol. 4, pp.131-146, 2003.
[38]: Subramanian et autres, 05 A. Suppapitnarm, A. Seffen, G.T. Parks and P.J. Clarkson, « A Simulated Annealing Algorithm for Multiobjective Optimisation », Engineering Optimization, Vol. 33, No. 1, pp. 59-85, 2000.
[39]: W. R Taylor, “Multiple sequence alignment by a pairwise algorithm”. *Comput Appl* Biosci, Vol. 3, pp. 81-7. 1987
[40]: Barton et Sternberg, 87 G. J. Barton, and M. J. Sternberg, “A strategy for the rapid multiple alignment of protein sequences. Confidence levels from tertiary structure comparisons”. *J. Mol Biol*, Vol. 198, pp. 327-37, 1987.
[41]: Thompson et autres, 94 J.D. Thompson, D.G. Higgins and T.J. Gibson, “CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment Through sequence weighting, position-specific gap penalties and weight matrix Choice”, *Nucleic Acids Res*. Vol. 22 No. 22 pp. 4673-4680, 1994.
[42]: Notredame et autres, 00 C. Notredame, L. Holm and D.G. Higgins, « Coffee: an objective function for multiple sequence alignments », *Bioinformatics*, Vol. 14, No. 5 pp. 407-422, 1998
[43]: Feng et Doolittle, 87: D.F. Feng and R.F Doolittle. “Progressive sequence Alignment as a prerequisite to correct phylogenetic trees”. *J. Mol. Evol*., Vol. 25, pp.351-360, 1987.
[44]: Thompson et autres, 94 J.D. Thompson, D.G. Higgins and T.J. Gibson, “CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment Through sequence weighting, position-specific gap penalties and weight matrix Choice”, *Nucleic Acids Res*. Vol. 22 No. 22 pp. 4673-4680, 1994
[45]: Saitou et Nei,87 N. Saitou, and M. Nei, “The neighbor-joining method: a new Method forreconstructing phylogenetic trees”. *Mol. Biol. Evol*., Vol. 4, pp. 406-425. 1987.
[46]: Edgar, 04 R.C Edgar, “MUSCLE: multiple sequence alignment with high accurac and high through put”, *Nucleic Acids Res*. Vol. 32, No. 5, pp. 1792-1797, 2004
[47]: J. Pei, R. Sadreyev and N.V. Grishin, “PCMA: fast and accurate multiple sequence Based profile consistency”, *Bioinformatics.* Vol. 19, pp. 427-428, 2003.
[48]: S.F. Altschul, T.L Madden, A.A. Schaffer, J. Zhang, Z. Zhang, Z. Miller, and D.J Lipman, “Gapped BLAST and PSIBLAST: à new generation of protein database search programs”. *Nucleic Acids Res.*, Vol. 25, pp. 3389–3402, 1997.
[49] :S. F. Altschul, W. Gish, W. Miller, E.W. Myers, D. J. Lipman,” Basic Local Alignment Search Tool”, *J. Mol. Biol.*, Vol. 215, pp. 403-410, 1990.
[50]: C.B. Do, M. Mahabhshyam, M. Brudno, and S. Batzoglou, “ProbCons: Probabilistic Consistency-based multiple sequence alignment”, *Genome res*. Vol. 15. pp. 330-340. 2005.
[51] C. Lambert, J. V. Campenhout, X. DeBolle and E. Depiereux, “Review of Common Sequence Alignment Methods: Clues to Enhance Reliability”, *Current Genomics*, vol. 4, pp. 131-146, 2003.
[52]: Lambert, 03 C. Lambert, J. V. Campenhout, X. DeBolle and E. Depiereux, “Review of Common Sequence Alignment Methods: Clues to Enhance Reliability”, *Current Genomics*, vol. 4, pp.131-146, 2003.
[53]: Tein et Ramli 2010, Layeb 2011, Gherboudj et al 2012
[54] : Yang et Deb, 2010

[55]: Cooren, 2008 Y. Cooren, A. Nakib, P. Siarry. Image Thresholding using TRIBES, a Parameter-free Particle Swarm Optimization Algorithm. Proceedings of the International Conference on Learning and Intelligent Optimization, pp 81-94. Springer, 2008.
[56]: Kennedy et Eberhart, 1995. J. Kennedy, R.C. Eberhart. A discrete binary version of the particle swarm algorithm. Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics, NJ: Piscatawary. pp. 4104-4109, 1997
[57] : Immuno-analyse & biologie specialisée. 2008 october; 23(5): 260-279
[58] : Immuno-analyse & biologie specialisée. 2008 october; 23(5): 260-279

Année universitaire : 2021-2022	**Présenté par :** **- HAFIDI MOUHAMED SAMER** **- DIOUANE LOUAI MOHAMED AZIZ**

Optimisation de l'alignement multiple des séquences avec la métaheuristique de recherche coucou

Mémoire pour l'obtention du diplôme de Master en : BIOINFORMATIQUE

La bioinformatique est une discipline qui vise le traitement automatique de l'information biologique. L'alignement multiple de séquences (MSA) constitue une tâche fondamentale pour beaucoup d'applications en bio-informatique y compris : la prédiction des structures primaires et secondaires des séquences, la détection de la distance phylogénétique, la prédiction des structures des molécules… etc.

Dans ce mémoire de fin d'étude, nous avons présenté les différentes méthodes d'alignement multiple des séquences. Ensuite, nous avons travaillé sur la métaheuristique nommée « Recherche Coucou » (en anglais : Cuckoo Search 'CS'). Pour cela, nous avons construit des fonctions pour adapter et utiliser l'algorithme CS pour l'alignement multiple des séquences. Les résultats obtenus ont été comparés avec ceux d'autres méthodes présentées dans la littérature. Cette comparaison a montré l'efficacité de la méthode proposée.

Mots-clefs : MSA : alignement multiple des séquences ; CS : coucou search ; PSO : Optimisation par essaim de particules ; CS MAS : Résultats de l'évaluation de la recherche Coucou.

Encadreur : DR. GHERBOUDJ AMIRA (MCA- Université Frères Mentouri, Constantine1).

Examinateur 1 : DR**.** DAAS Mohamed Skander **(**MCA - Université Frères Mentouri, Constantine 1**).**

Examinateur 2 : DR**.** BELLIL Ines **(**MCA - Université Frères Mentouri, Constantine 1).