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 1
Frères Mentouri Constantine 1 University
Université Frères Mentouri Constantine 1

Faculté des Sciences de la Nature et de la Vie — **كلية علوم الطبيعة والحياة**
Département de Biologie Appliquée — **قسم البيولوجيا التطبيقية**

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

N° d'ordre :
N° de série :

Intitulé :

Méthode Computationnelle basée sur l'optimisation par essaim de particule pour alignement multiple des séquences

Présenté par : MAKOUF Amir
REBOUH Mounder

Le 30/06/2022

Jury d'évaluation :

Encadreur : Dr. Amira GHERBOUDJ ; MCA - Université Frères Mentouri, Constantine 1.
Examinateur 1: Pr. Abdelhafid HAMIDECHI **;** Pr - Université Frères Mentouri, Constantine 1.
Examinateur 2 : Dr. Hamza CHEHILI ; MCA - Université Frères Mentouri, Constantine 1.

Année universitaire : 2021-2022

Remerciements

En premier lieu, nous tenons à remercier notre DIEU, notre créateur pour nous avoir donné la force pour accomplir ce travail.

Nous tenons à exprimer nos vifs remerciements à tous les professeurs qui nous ont aidés tout au long de notre cursus universitaire en particulier notre encadrante :

M*^me Dr^ *GHERBOUDJ Amira

Pour ses conseils et l'aide qu'il nous a apportés.

Nous remercions vivement les membres de jury qui nous ont fait l'honneur de juger ce travail, notamment :

Pr Abdelhafid HAMIDECHI …… (Examinateur 1)

***D*r Hamza CHEHILI ……. (Examinateur 2)**

Nos derniers remerciements, vont à tous ceux qui ont contribué de près ou de loin pour l'aboutissement de ce travail.

Dédicace

Je dédie ce travail à :

A mes chers parents, ma mère Derouiche Farida et mon père Makouf Mohamed
Pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,
A mes chères sœurs Dr Bessma ; Dr Rayane et Dr Lina

Pour leurs encouragements permanents, et leur soutien moral,
A mon cher petit frère Wassim

À mes chers collègues : Guerroudj Roumaissa, Rebouh Mounder

A toute ma famille pour leur soutien tout au long de mon parcours universitaire.

A tous mes amis de près ou de loin.

A tous ceux qui m'ont toujours soutenu, supporter et était toujours présent, même dans les pires moments.

Je vous dédie ce travail tout en espérant le succès dans votre vie familiale et professionnelle.

AMIR MAKOUF

Dédicace

Je dédie ce modeste travail à mes chers parents qui m'ont donné une éducation digne, à toute ma famille, leur amour a fait de moi ce que je suis aujourd'hui,

A ma sœur qui est loin des yeux mais près du cœur

A mon frère DR. REZGUI ABDELMALEK

A tous mes Professeurs, mes Collègues en particulier AMIR MAKOUF

A tous mes amis

MERCI

MOUNDER REBOUH

Liste des abréviations

ADN	Acide Désoxyribonucléique.
ARN	Acide ribonucléique.
L'ARNm	L'acide ribonucléique messager.
AA	Les acides aminés.
NGS	Séquençage nouvelle génération.
PCR	Polymérase Chain Réaction.
Indel /Gap	Décalage entre deux particules (Insertion/Délétion)
MSA	Alignement Multiple de Séquences
IA	Intelligence Artificielle
RO	Recherche Opérationnelle
B&B	Branch and Bound.
MAFFT	Alignement multiple à l'aide de la transformation de Fourier rapide
UPGMA	Méthode de groupe de paires non pondérée avec moyenne arithmétique
T-Coffee	Fonction d'objectif de consistance basée sur l'arborescence pour l'évaluation de l'alignement.
NJ	Neighbor Jining.
PCMA	Profile Consistance Multiple Séquence Alignement.
COMPASS	Comparaison de plusieurs alignements de protéines avec évaluation de la signification statistique.
PSO	Optimisation Par essaims de Particules.

Liste des figures

Liste des tableaux

Table des Matières

Chapitre 2 : Les Méthodes D'Alignement Multiple Des Séquences

Chapitre 3 : Algorithmes d'optimisation par essaims particulaires

Chapitre 4 : Implémentation et discussion

Résumé

En bio-informatique, l'alignement des séquences est une méthode consistant à représenter deux ou plusieurs séquences de macromolécules biologiques (ADN, ARN ou protéines) pour identifier des régions de similarité qui peuvent être fonctionnelles, structurelles ou évolutives entre les séquences, elles sont considérées comme une partie fondamentale des processus d'une multitude d'applications dans ce domaine qui sert à traiter automatiquement l'information biologique.

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 « optimisation par essaim de particules » (en anglais : Particle Swarm Optimization : PSO). Pour cela, nous avons construit des fonctions pour adapter et utiliser l'algorithme PSO 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 : Bio-informatique, Alignements Multiple de Séquence, Metaheuristique, Algorithme PSO

Abstract

In bioinformatics, sequence alignment is a method of representing two or more sequences of biological macromolecules (DNA, RNA or proteins) to identify regions of similarity that may be functional, structural or evolutionary between the sequences, they are considered as a fundamental part of the processes of a multitude of applications in this field, which serves to automatically process biological information.

In this dissertation, we presented the different methods of multiple sequence alignment. Then, we worked on the metaheuristic called “particle swarm optimization” (in English: Particle Swarm Optimization: PSO). For this, we have built functions to adapt and use the PSO algorithm for multiple sequence alignment. The results obtained were compared with those of other methods presented in the literature. This comparison showed the effectiveness of the proposed method.

Keywords: Bioinformatics, Multiple Sequence Alignment, Metaheuristics, Algorithm Particle Swarm Optimization.

ملخص

في المعلوماتية الحيوية، تعد محاذاة التسلسل طريقة لرسم خرائط اثنين أو أكثر من تسلسل الجزيئات البيولوجية لتحديد مناطق التشابه التي قد تكون وظيفية أو هيكلية أو تطورية بين التسلسلات، (أو البروتينات RNA أو DNA) وتعتبر جزءًا أساسيًا من عمليات العديد من التطبيقات في هذا المجال والتي تعمل على معالجة المعلومات البيولوجية تلقائيًا. في هذه الرسالة، قدمنا الطرق المختلفة لمحاذاة التسلسل المتعدد. بعد ذلك، عملنا على metaheuristic المسمى "تحسين سرب الجسيمات". لهذا، قمنا ببناء وظائف لتكييف واستخدام خوارزمية PSO لمحاذاة التسلسل المتعدد. تمت مقارنة النتائج التي تم الحصول عليها مع تلك الخاصة بالطرق الأخرى المعروضة في الأدبيات. أظهرت هذه المقارنة فعالية الطريقة المقترحة.

الكلمات الدالة: المعلوماتية الحيوية، محاذاة التسلسل المتعددة، الاستدلال الفوقي، خوارزمية تحسين سرب الجسيمات.

Introduction Générale

Une séquence génomique est l'enchaînement des nucléotides le long d'une macromolécule d'ADN. Elle peut être représentée par une chaîne de caractères utilisant l'alphabet des quatre lettres A, C, G et T, initiales des bases azotées – Adénine, Cytosine, Guanine et Thymine – qui distinguent les quatre types de nucléotides. C'est l'enchaînement des nucléotides au sein des régions codantes des gènes qui dicte la suite des acides aminés qui compose un polypeptide, dont le repliement et diverses modifications chimiques conduiront à une protéine fonctionnelle. Une séquence protéique est l'enchaînement des vingt types d'acides aminés le long d'un polypeptide. Cette séquence est classiquement représentée par une chaîne de caractères qui utilise un alphabet de vingt lettres [1].

L'alignement des séquences génomiques et protéiques est la tâche informatique la plus exécutée par les biologistes. Des algorithmes sont mis en œuvre pour calculer les meilleurs alignements entre plusieurs séquences [1]. Il s'agit de déterminer dans quelle mesure deux séquences, génomiques ou protéiques, se ressemblent [1]. Il permet de :

- Détecter des résidus identiques ou similaires pouvant jouer un rôle clé dans la fonction de la molécule ou dans sa structure tridimensionnelle.
- Caractériser de nouvelles familles de protéines.
- Détecter ou démontrer une homologie entre différentes séquences
- Trouver un PRIMER consensus pour des PCR
- Etablir une phylogénie
- Aider à la modélisation : les algorithmes de prédiction de structure secondaire exploitent très bien les alignements multiples.
- Le traitement est très long et dépend de trois paramètres : le volume de données à traiter, la puissance de calcul des ordinateurs et les algorithmes utilisés.

Depuis les années 70, des méthodes algorithmiques exactes basées sur la programmation dynamique ont été proposées afin d'automatiser la tâche de l'alignement des séquences. Cependant, la recherche d'un alignement des séquences exacte implique souvent une exploration très vaste de l'espaces de recherche et dont la complexité devient de plus en plus critique avec le nombre et la taille des séquences à aligner.

L'alignement multiple des séquences (en l'anglais : Multiple Sequence Alignement : MSA) a été démontré comme un problème NP-complet dont la complexité augmente avec la taille de l'instance à aligner. Raison pour laquelle, il ne peut être résolu par une méthode exacte que

pour des séquences de petites tailles. Il a donc fallu trouver des moyens rapides pour pouvoir trouver des solutions réalisables avec des couts de réponse raisonnables.

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 CLUSTALW [2], les méthodes itératives comme la métaheuristique MAAFT [3], les méthodes basées sur la consistance comme la méthode PCMA [4] et les méthodes évolutionnaires comme la métaheuristique dites algorithme génétique [5]. L'alignement des séquences est utilisé pour comparer des séquences homologues, de longueur comparable, et aussi comparer un gène et l'ARNm qui en est issu pour mettre en évidence la structure morcelée des gènes d'eucaryotes et repérer exons et introns. [6]

L'objectif de PSO est trouvé l'optimum global de quelque multidimensionnel (habituellement non linéaire), l'algorithme a prouvé son efficacité et a résolu beaucoup de problèmes [7].

Dans ce mémoire, nous présentons une utilisation avec adaptation de la métaheuristique d'optimisation par essaim de particules (en anglais : Particle Swarm Optimization : PSO) pour résolution du problème MAS. PSO est une des méthodes d'optimisation. Il s'agit d'une métaheuristique qui s'inspire de l'intelligence par essaim, une catégorie alternative de la catégorie des algorithmes évolutionnaires.

Notre mémoire est organisé en quatre chapitres :

- Dans le premier chapitre, nous présentons les concepts de base sur la biologie moléculaire dans le premier chapitre.
- Le deuxième chapitre décrit principalement les différents types de méthodes d'alignement multi-séquences et leurs différences.
- Dans le troisième chapitre, nous donnons une explication générale de l'algorithme PSO.
- Le chapitre 4 est consacrée à la présentation de notre travail qui consiste à montrer l'efficacité des méthodes approchées pour le problème d'alignement multiple de séquences ou la méta-heuristique PSO sera formulée, implémentée et évaluée pour un tel problème.

Chapitre 01:

La Biologie Moléculaire

Chapitre 01 : La Biologie Moléculaire

1. Introduction

La biologie moléculaire (parfois abrégée bio mol ou BM) est une discipline scientifique au croisement de la génétique, de la biochimie et de la physique, dont l'objet est la compréhension des mécanismes de fonctionnement de la cellule au niveau moléculaire. Le terme « biologie moléculaire », utilisé la première fois en 1938 par Warren Weaver, désigne également l'ensemble des techniques de manipulation d'acides nucléiques (ADN, ARN), appelées aussi techniques de génie génétique [8]. Désigne également l'ensemble des stratégies de manipulation d'acides nucléiques (ADN, ARN), appelées aussi stratégies de génie génétique.

La bio-informatique est l'appellation donnée à l'application des technologies informatiques au domaine de la biologie. Elle vise à accroître la compréhension des processus biologiques et à en modéliser certaines facettes pour simuler des comportements ou des phénomènes porteurs de sens. Elle se distingue des autres approches par l'accent qu'elle met sur le développement et l'application des techniques de calcul intensives telles que la reconnaissance des formes, l'exploration de données et les algorithmes d'apprentissage automatique pour atteindre ses objectifs [9]**.**

La bio-informatique a pour but d'étudier la composition biologique et le développement des molécules, des cellules, des tissus et des organismes ainsi que leur structure et leur fonctionnement.

2. La cellule vivante

- La cellule est une unité fondamentale, structurale et fonctionnelle des organismes vivants. Elle peut remplir toutes les fonctions de l'organisme, à savoir le métabolisme, le mouvement, la croissance, la reproduction ou encore la transmission de gènes.

- C'est une entité vivante qui fonctionne de manière autonome, tout en restant coordonnée avec les autres [10] .

- Tout organisme vivant est composé de cellule.

- Il y a des organismes unicellulaires (bactéries, levure...) ou multicellulaires.
- Une cellule est une solution contenant différentes molécules entourées d'une membrane.

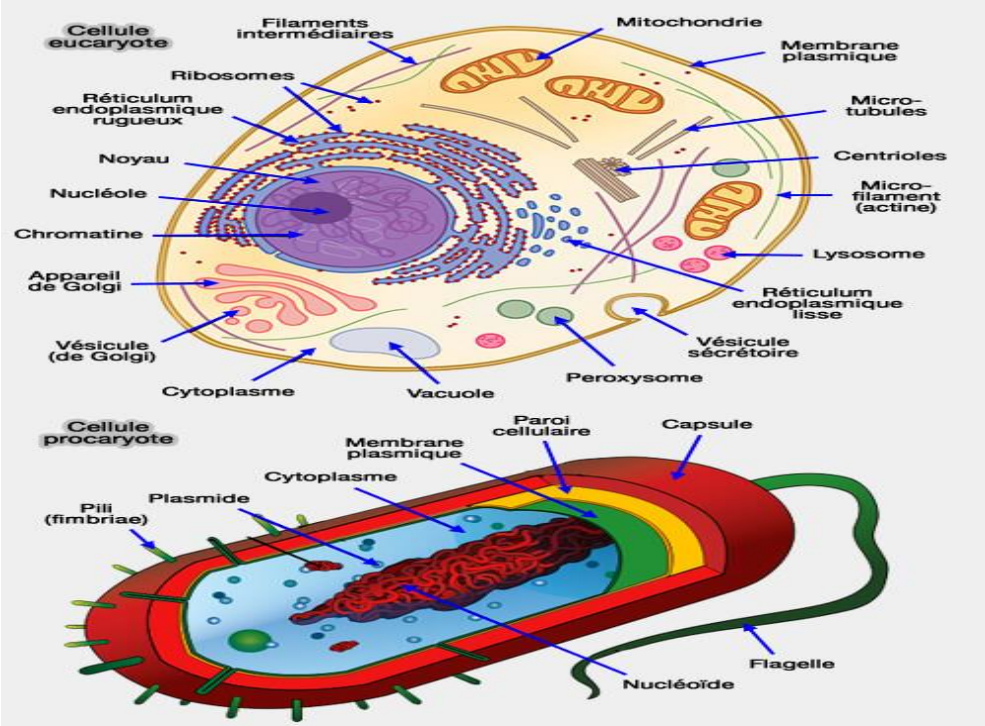


Figure n°01 : Cellules eucaryote et procaryote. [10]

3. La vie cellulaire

L'information génétique est stockée de manière identique dans chaque cellule sous la forme de polymères connus sous le nom d'acides nucléiques. Les acides nucléiques existent sous deux formes, l'acide désoxyribonucléique ou ADN, et l'acide ribonucléique ou ARN.

Le fait que l'ADN soit le détenteur de l'information génétique a été mis en évidence dans les années 40 par Oswald Avery. Formalise la transformation de l'information génétique en molécules fonctionnelles. Cette information transite majoritairement de l'ADN, dépositaire du code génétique, aux protéines, qui sont les constituants élémentaires servant au fonctionnement de la cellule et de l'organisme entier. L'ARN joue dans ce schéma le rôle d'une molécule transitoire assurant la migration du code génétique vers une machine de traduction en protéines [11] .

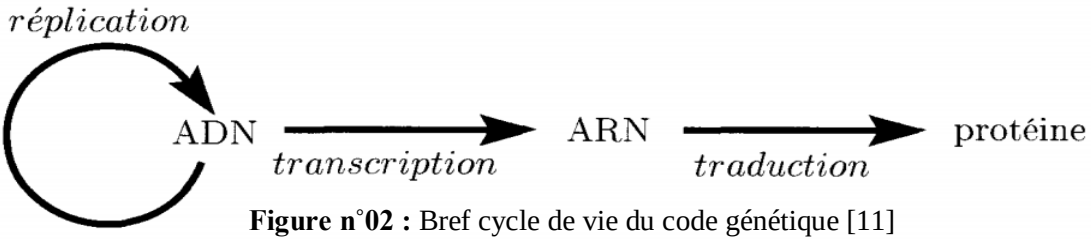


Figure n°02 : Bref cycle de vie du code génétique [11]

4. Les molécules de la vie

On les regroupe en 4 grandes familles :

- ✓ **Les petites molécules**
- ✓ **Les protéines**
- ✓ **L'ADN**
- ✓ **L'ARN**

4.1. Les petites molécules

- Les petites molécules ayant un rôle : ATP, NADPH stockent l'énergie.
- Sucres, lipides (sources d'énergie, structure des membranes).
- Acides aminés et nucléotides, qui sont les blocs de base pour former les protéines et l'ADN/l'ARN [12].

4.2. Les protéines

Une protéine est une macromolécule biologique composée par une ou plusieurs chaînes d'acides aminés liés entre eux par des liaisons peptidiques. En général, on parle de protéine lorsque la chaîne contient plus de 50 AA, pour des tailles plus petites, on parle de peptide et de polypeptide.

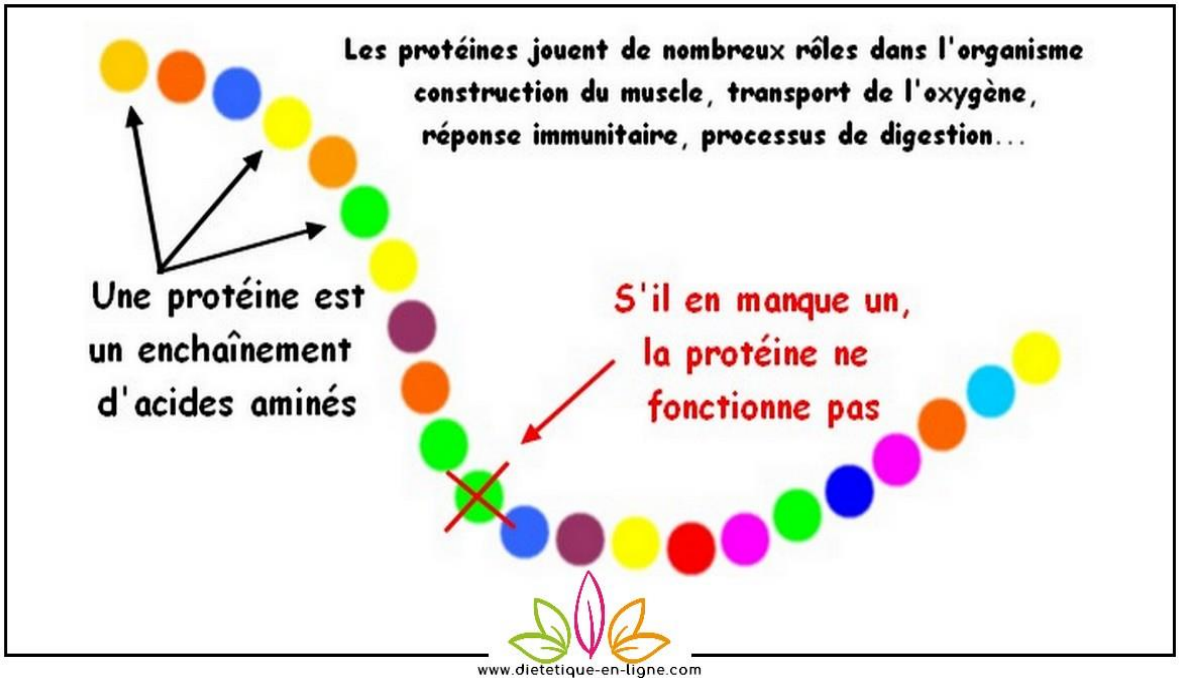


Figure n°03 : Protéine = polymère d'AA [13]

4.2.1. Structure primaire et variabilité des protéines

La structure primaire des protéines est représentée par la séquence d'acides aminés qui se lient de manière à former une chaîne polypeptidique.

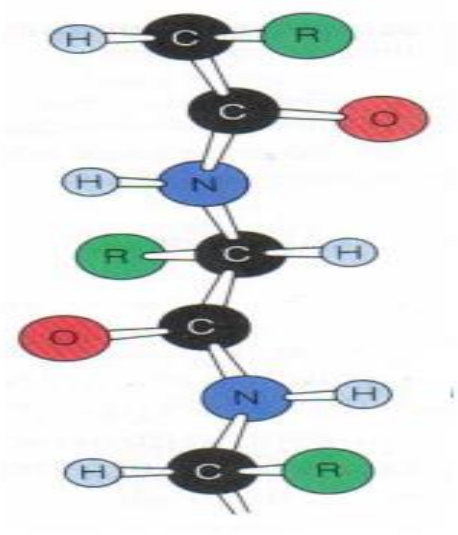


Figure n°04 : Structure primaire d'une protéine [13].

Les propriétés uniques de chaque protéine dépendent des types d'acides aminés qui la composent et de leur séquence, on peut créer une nouvelle protéine de fonction différente en remplaçant un acide aminé ou en changeant sa position. Tout comme il arrive que les changements de la combinaison des acides aminés donnent des protéines non fonctionnelles. Exemple : les hémoglobines pathologiques humaines (hémoglobines falciforme S et anémiante C) ne diffèrent de l'hémoglobine normale qu'au niveau du sixième résidu de la chaîne β (remplacement de l'acide glutamique respectivement par la valine et la lysine) [13].

4.2.2. Structure tridimensionnelle des protéines

1. Structure secondaire

Les protéines n'existent pas sous forme de chaînes linéaires d'acides aminés : elles se tordent et se replient sur elles-mêmes. C'est leur structure secondaire. La structure secondaire la plus courante est celle de l'hélice alpha (α). Dans l'hélice alpha, la chaîne primaire s'enroule sur elle-même puis est stabilisée par des liaisons hydrogène entre les groupes NH et CO, à tous les quatre acides aminés environ [13].

Le feuillet plissé bêta (β) est une autre structure secondaire, où les chaînes polypeptidiques primaires ne s'enroulent pas mais se lient côte à côte au moyen de liaisons hydrogène et forment une sorte d'échelle pliante (fig5). Dans ce type de structure secondaire, les liaisons hydrogène

peuvent unir différentes parties d'une même chaîne qui s'est repliée sur elle-même en accordéon ou encore différentes chaînes polypeptidiques. Dans les hélices alpha, les liaisons hydrogène unissent toujours différentes parties d'une même chaîne. Une chaîne polypeptidique peut présenter les deux types de structure secondaire [13].

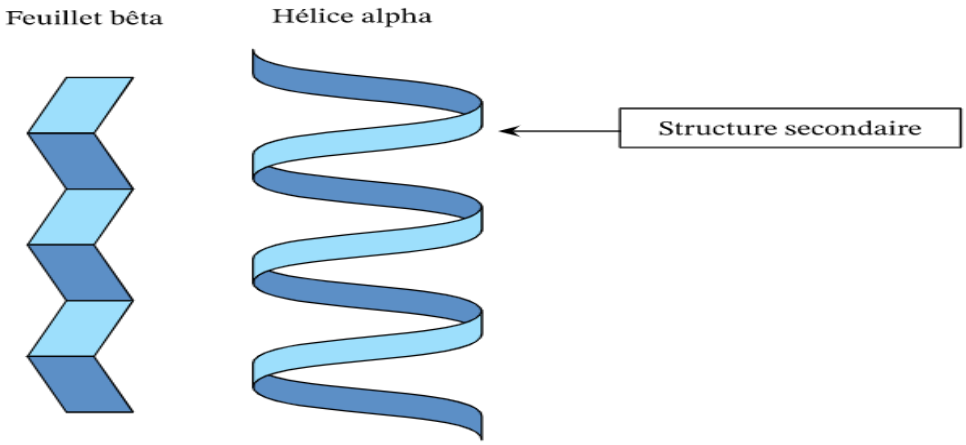


Figure n°05 : Schéma montrant l'hélice α et le feuillet β que peuvent être formés dans la structure secondaire d'une protéine. [13]

2. Structure tertiaire et quaternaire

Un grand nombre de protéines se complexifient jusqu'à la structure tertiaire, une structure très spécifique formée à partir de la structure secondaire. Dans une structure tertiaire, des régions hélicoïdales ou plissées de la chaîne polypeptidique se replient les unes sur les autres et forment une molécule en forme de boule, ou molécule globulaire. La structure tertiaire est maintenue par des liaisons (covalentes, hydrogène ...) entre des acides aminés souvent très éloignés sur la chaîne primaire (fig6-a).

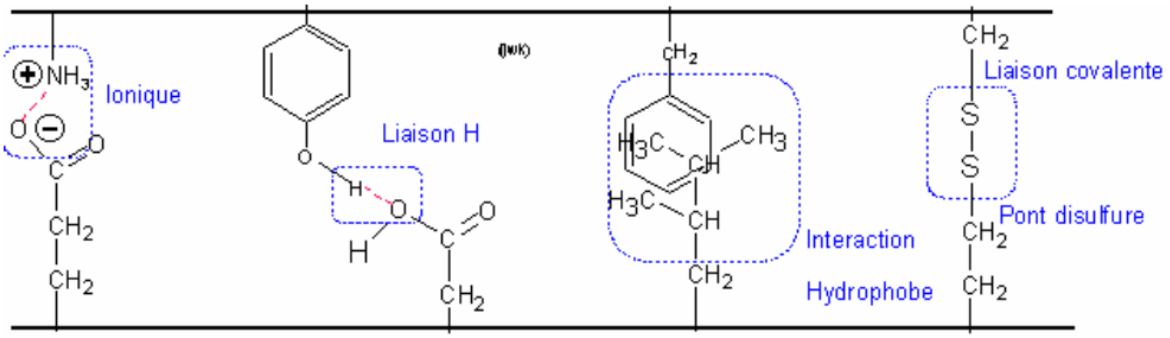


Figure n°06 (a) : Liaisons secondaires interatomiques dans une protéine. [13]

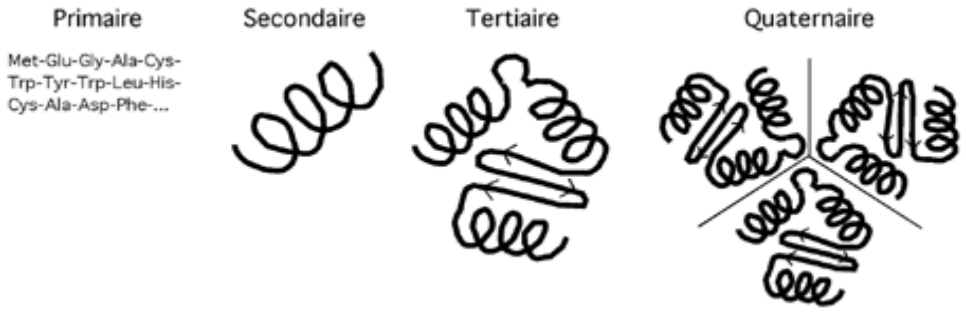


Figure n°06 (b) : Les différentes structures de protéine [13]

4 .3. L'ADN et l'ARN

L'ADN est une molécule très longue et c'est le support de l'information génétique héréditaire, composée d'une succession de nucléotides accrochés les uns aux autres par des liaisons phospho diester. Il existe quatre nucléotides différents : l'adénosine, la cytosine, la guanine et la thymine, dont l'ordre d'enchaînement est très précis et correspond à l'information génétique.

L'ARNm est le résultat de la transcription de la séquence d'ADN d'un gène. Une fois que la séquence est transcrite, elle devient en fait un code composé d'une série de triplets de nucléotides appelés codons. Chaque codon est associé à l'un des 20 AA utilisés pour la synthèse des protéines, mais un même acide aminé peut être codé par plusieurs codons [14].

L'ADN (acide désoxyribonucléique) et l'ARN (acide ribonucléique) sont deux molécules légèrement différentes. D'abord, le sucre qui compose les nucléotides de l'ARN est un ribose, alors que celui des nucléotides de l'ADN est un désoxyribose, d'où la différence de nom entre les deux molécules. Aussi, la thymine (T) présente dans l'ADN est remplacée par l'uracile (U) dans l'ARN [14]. Finalement, l'ARN se trouve le plus souvent dans les cellules sous forme monocaténaire, c'est-à-dire de simple brin, comparativement à l'ADN qui est formé de deux brins complémentaires enroulés en hélice (double hélice).

Trois différences structurales distinguent l'ADN de l'ARN. Elles sont décrites dans le tableau suivant :

Le tableau 1 : La différence entre l'ADN et l'ARN

Caractéristiques	ADN	ARN
Nom du sucre utilisé	Désoxyribose (Pour le « D » ADN)	Ribose (pour le « R » ARN)
Noms des quatre bases utilisées	A (Adénine)	A (Adénine)
	T (Thymine)	U (Uracile)
	C (Cytosine)	C (Cytosine)
	G (Guanine)	G (Guanine)
Nombre de chaînes	2	1

4.4. La relation entre ADN et protéine

Les gènes sont des segments de la molécule d'ADN codant pour des protéines. La séquence des nucléotides dans l'ADN gouverne la séquence des acides aminés dans la protéine selon un système de correspondance : le code génétique [15] .

5. La structure de l'ARN

Les protéines sont par excellence les molécules fonctionnelles de l'organisme. Mais certains ARN sont eux aussi directement fonctionnels, les ARN de transfert et ribosomique, ainsi que de nombreuses autres familles. Leur fonction, comme pour les protéines, est liée à leur structure, qui peut elle-aussi être décrite de façon hiérarchique : structure primaire, secondaire, tertiaire, et tridimensionnelle. Nous examinerons plus particulièrement la structure secondaire de l'ARN [16] . Parce que c'est ce qui m'intéresse le plus dans cette recherche.

La structure secondaire : d'une molécule d'ARN est l'ensemble des tiges qui se forment, pourvu qu'elles respectent les contraintes suivantes : on impose qu'elles soient emboîtées ou juxtaposées, comme sur la (Fig.07-a). Dans la formation des tiges il peut malgré tout survenir des croisements (Fig.07-b). De tels croisements sont appelés des pseudos nœuds car jusqu'à présent jamais aucun nœud n'a été observé sur des molécules d'ARN. Par ailleurs le diagramme ne permet pas de distinguer si le squelette de la molécule est noué ou non. A cause de sa

structure en hélice de type A (11 paires par tour), l'ARN ne peut pas former des pseudos nœuds dont les tiges excèdent 9 à 10 paires de bases. Cette contrainte restreint considérablement leur occurrence [17] .

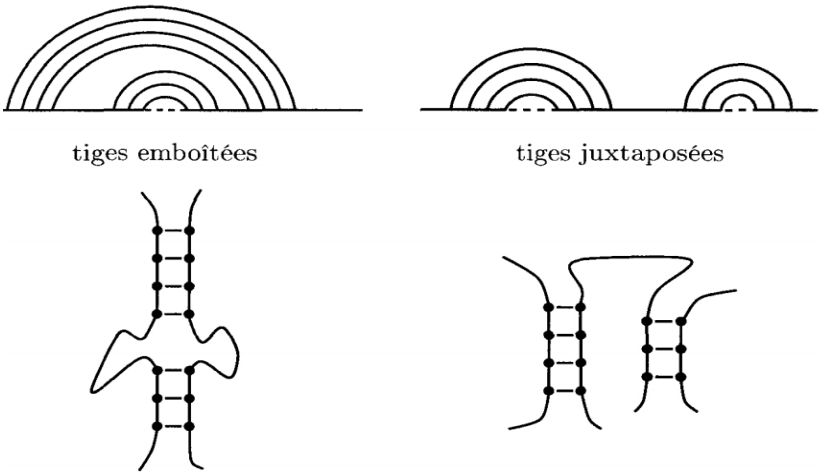


Figure n°07 (a) : La structure secondaire est constituée de tiges emboîtées ou juxtaposées.[16]

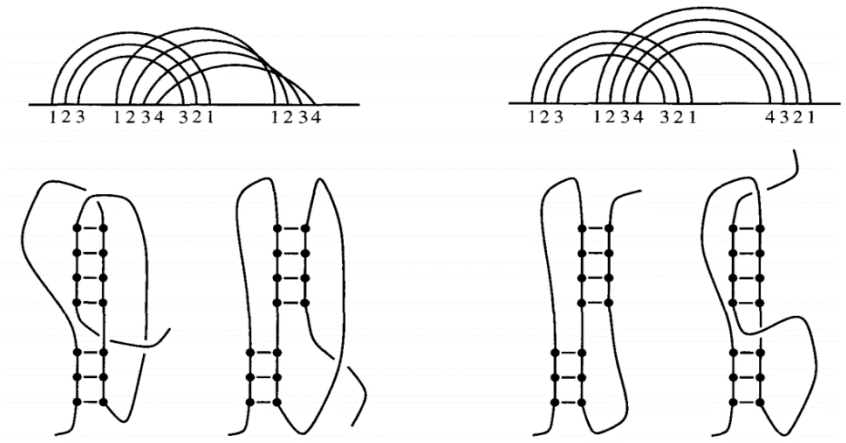


Figure n°07 (b) : Pseudo nœuds. [16]

L'empilement des appariements pour former des tiges donne à la structure secondaire un caractère modulaire simple qui permet de la décrire comme un assemblage de différentes structures élémentaires [16] .

5.1. Détermination de la structure secondaire

La structure secondaire de l'ARN peut être déterminée à partir des coordonnées atomiques (structure tertiaire) obtenues par cristallographie aux rayons X, souvent déposées dans la banque de données sur les protéines [18] [19] ; Les méthodes actuelles incluent 3DNA.

5.2. Comparaison des ARN

La comparaison de structures secondaires d'ARN trouve de nombreuses applications en biologie. En effet, il est couramment admis que si deux ARN ont une structure spatiale proche, alors ils ont la même fonction biologique [20].

5.3. Classification

Le problème de la classification ("clustering") est de partitionner un ensemble de structures secondaires selon leur similarité. Pour cela, on dispose d'un ensemble de structures dont on ne connait pas a priori la fonction. Dans un premier temps, on compare chacune de ces structures deux a deux. Puis, on constitue des groupes de telle façon à ce que toutes les structures secondaires au sein d'un même groupe soient "proches" [21] .L'une des difficultés est de définir la notion de "proche" et d´établir le nombre de groupes "idéal". En effet, une solution triviale à ce problème revient à créer autant de groupes qu'il y a de séquences. La solution opposée consiste à n'avoir qu'un seul groupe pour l'ensemble des séquences. Ainsi, on voit bien qu'il faut mettre en place des critères à la fois sur le nombre de groupes et la similarité des structures au sein de chaque groupe. L'outil utilisé pour la comparaison des structures tient une place importante dans cette démarche car il sert à établir la similarité entre les ARN.

6. Le séquençage d'ADN

Le séquençage de l'ADN constitue une méthode dont le but est de déterminer la succession linéaire des bases A, C, G et T prenant part à la structure de l'ADN. La lecture de cette séquence permet d'étudier l'information biologique contenue par celle-ci. Étant donné l'unicité et la spécificité de la structure de l'ADN chez chaque individu, la séquence de l'ADN permet de nombreuses applications dans le domaine de la médecine, comme, par exemple, le diagnostic, les études génétiques, l'étude de paternité, la criminologie, la compréhension de mécanismes physiopathologiques, la synthèse de médicaments, les enquêtes épidémiologiques [22] .

Dans les études de génomes, le terme de re-séquençage (expression pouvant prêter à confusion) est utilisé à la place de séquençage. Cette dénomination, essentiellement utilisée en génétique, désigne le séquençage d'un segment d'ADN suivi de la comparaison du résultat obtenu avec celui d'une séquence de référence connue. Un autre terme est également employé : le

séquençage de novo. Dans ce cas, il s'agit du séquençage d'un génome pour lequel il n'existe pas de séquence référence. Il s'agit donc de la détermination d'une séquence inconnue.

Les deux premières techniques de séquençage de l'ADN, celle de Maxam-Gilbert [23] et celle de Sanger [24] ont été décrites en 1977.Il s'agissait du séquençage de l'opérateur Lac et de l'ARNm de celui-ci. La technique de Sanger a révolutionné le monde de la biologie moléculaire en permettant de décrypter différents génomes, tel que celui du génome humain complètement déchiffré en 2006 ou d'autres génomes, le génome bactérien, par exemple, le premier d'entre eux étant celui d'Haemophilus influenzae, complètement décrit en 1995 [25]. Bien que les techniques de séquençage évoluent, comme nous allons le voir dans cet article, la méthode de Sanger continue d'être la méthode la plus employée dans le monde à l'heure actuelle.

La technique de Maxam-Gilbert Cette technique est pratiquement abandonnée de nos jours [22].

- **Le NGS :**

La commercialisation depuis 2005 des technologies de NGS a révolutionné au cours de ces dernières années la dimension des analyses génétiques par un changement majeur d'échelle des capacités de séquençage [26] .

C'est technologie récente permettent de séquencer l'ADN et l'ARN beaucoup plus rapidement que les méthodes précédentes comme le séquençage de Sanger, et comme tels ont révolutionné l'étude de la génomique et de la biologie moléculaire [27].

La technologie NGS présentent 3 étapes communes :

1. La préparation de banques : les banques sont créées en utilisant une fragmentation
2. L'amplification : la banque est amplifiée grâce à des méthodes d'amplification clonale et de PCR.
3. Le séquençage : l'ADN est séquencé en utilisant différentes approches en fonction de la technologie utilisée.

7. L'Assemblage

En bio-informatique, l'assemblage consiste à aligner et/ou fusionner des fragments d'ADN ou d'ARN issus d'une plus longue séquence afin de reconstruire la séquence originale. Il s'agit d'une étape d'analyse in silico qui succède au séquençage de l'ADN ou de l'ARN d'un organisme unique, d'une colonie de clones (bactériens par exemple), ou encore d'un mélange complexe d'organismes.

Le problème de l'assemblage peut être comparé à celui de la reconstruction du texte d'un livre à partir de plusieurs copies de celui-ci, préalablement déchiquetées en petits morceaux.

Les stratégies d'assemblage peuvent être organisées en 3 principaux paradigmes [28] :

1. Glouton.
2. Overlap-Layout-Consensus (OLC).
3. Graphe de De Bruijn.

Les performances des technologies NGS reposent en grande partie sur les méthodologies et les capacités bio-informatiques de reconstituer l'intégralité des séquences visées (contigs ou scaffolds) à partir d'un très grand nombre de fragments de génomes ou transcriptomes.

La première tâche est d'ordonner et d'assembler ces fragments d'information en s'appuyant sur des génomes de référence. Le but est d'identifier les variations avec le standard, puis, dans un second temps, d'en comprendre le rôle et l'influence (commentaire, interprétation) [29].

8. Alignement des séquences biologique

8.1. Données biologiques

- Séquence génomique = suite de lettres

  - Séquence nucléotidique (ADN) : 4 acides nucléiques

ATGAAGGCTCCCACCGTGCTGGCACCTGGCATTCTGGTGCTGCTGCTTGTCCTTG-

  - Séquence protéique (protéine) : 20 AA

MKAPTVLAPGILVLLLSLVQRSHGECKEALVKSEMNVNMKYQLPNFTAET

Comparaison de séquences

Étude de l'histoire évolutive des séquences

Etude de l'évolution des organismes vivants

8.2. Alignement

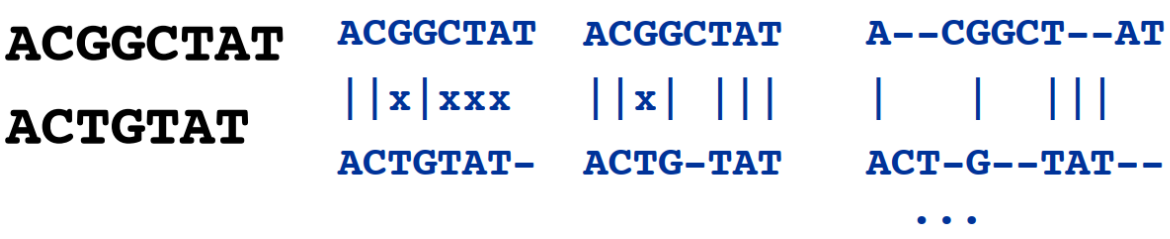
- Mise en correspondance de deux séquences (ADN ou Protéines) pour faire apparaître les similarités, i.e., segments communs.
- Aligner deux séquences globalement c'est les réécrire :
  - ✓ On rajoute des caractères de gap "-".
  - ✓ Elles font la même longueur.
  - ✓ On ne met pas de gap en face l'un de l'autre.

8.3. Score d'un alignement

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. Il tient compte d'une part du nombre d'acide aminés identiques entre les deux séquences et d'autre part du nombre d'acides aminés similaires sur le plan physico-chimique [30] .

On doit ordonner les qualités avec un score et on somme 3 événements élémentaires le long de l'alignement :

- ✓ Même lettre : match
- ✓ Lettre différente : mismatch
- ✓ Insertion/Délétion (indel)



8.4. Pourquoi aligner des séquences

L'objectif de l'alignement est de disposer les composants (nucléotides ou aa) et évaluation de la ressemblance globale entre deux séquences pour identifier les zones de concordance. Ces

alignements sont réalisés par des programmes informatiques dont l'objectif est de maximiser le nombre de coïncidences entre nucléotides ou acides aminés dans les différentes séquences.

8.5. Type d'alignements

On distingue 2 types d'alignements qui diffèrent suivant leur complexité :

- **L'alignement par paires** : consiste à aligner 2 séquences peut être réalisé grâce à un algorithme de complexité polynomiale. Il est possible de réaliser un alignement :
  - ✓ **Global**, c'est à dire entre les 2 séquences sur toutes leurs longueurs
  - ✓ **Local** entre une séquence et une partie de l'autre séquence
- **L'alignement multiple**, qui est un alignement global : consiste à aligner plus de 2 séquences et nécessite un temps de calcul et un espace de stockage exponentiel en fonction de la taille des données.

9. Prédiction de la structure secondaire

La plupart des méthodes de prédiction de la structure secondaire des acides nucléiques reposent sur un modèle thermodynamique voisin le plus proche [31] [32]. Une méthode courante pour déterminer les structures les plus probables étant donné une séquence de nucléotides utilise un algorithme de programmation dynamique qui cherche à trouver des structures à faible énergie libre. Les algorithmes de programmation dynamique interdisent souvent les pseudoknots, ou d'autres cas dans lesquels les paires de bases ne sont pas complètement imbriquées, car la prise en compte de ces structures devient très coûteuse en calcul, même pour les petites molécules d'acide nucléique. D'autres méthodes, telles que les grammaires stochastiques sans contexte, peuvent également être utilisées pour prédire la structure secondaire de l'acide nucléique [33]. Pour de nombreuses molécules d'ARN, la structure secondaire est très importante pour le bon fonctionnement de l'ARN - souvent plus que la séquence réelle. Ce fait facilite l'analyse de l'ARN non codant parfois appelé « gènes ARN ». Une application de la bio-informatique utilise des structures secondaires d'ARN prédites dans la recherche d'un génome pour des formes non codantes mais fonctionnelles d'ARN. Par exemple, les micro-ARN ont des structures canoniques longues tige-boucle interrompues par de petites boucles internes [34].

Chapitre 02:

Les Méthodes D'Alignement Multiple Des Séquences

Chapitre 02 : Les Méthodes D'Alignement Multiple Des Séquences

1. Introduction

L'alignement multiple de séquences MSA (Multiple Séquence Alignement) 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. C'est un problème important en bio-informatique, il permet de déterminer des relations génétiques et phylogénétiques [35].

Le MSA peut être considérer comme problème d'optimisation combinatoire car il possède généralement un nombre énorme de solutions réalisables. La résolution la plus évidente est de lister toutes les combinaisons possibles afin de trouver celles qui sont valides et meilleures. Ce type d'approche utilisent algorithmes Exhaustifs. Ces dernières sont très gourmandes en termes de complexité. Ils passent d'algorithmes polynomiaux à non-polynomiaux avec l'augmentation de la dimension du problème à traiter.

Pour faire face à sa complexité, le MSA a été traité par des méthodes inspirées de l'intelligence artificielle et la recherche opérationnelle telles que les méthodes d'optimisation combinatoire. Dans ce chapitre, nous présentant les méthodes d'alignement multiple de séquence les plus connues dans la littérature.

2. Intelligence Artificielle

L'Intelligence Artificielle (abrégée IA) apparue en 1956 est la science dont le but est de faire par une machine des tâches que l'homme accomplit en utilisant son intelligence [36]

L'Informatique est la science du traitement de l'Information, l'IA s'intéresse à tous les cas où ce traitement ne peut être ramené à une méthode simple, précise, algorithmique. Un algorithme est une suite d'opérations ordonnées, bien définies, exécutables sur un ordinateur actuel, et qui permet d'arriver à la solution en un temps raisonnable (minutes, heures, ou plus, ... mais pas des siècles) [36].

Intelligence Artificielle et Biologie médicale, Aide au diagnostic, choix du traitement le plus adapté et autres prouesses, l'intelligence artificielle s'impose depuis plusieurs années dans le domaine médicale et n'a de cesse d'étonner.

Alors la technologie est-elle sur la voie de remplacer l'humain ? Non affirment les professionnels des domaines médicales. Moins qu'un concurrent, l'IA est avant tout un outil efficace, une aide avant tout. Et cela se vérifie dans le domaine de la biologie médicale dans lequel l'IA permet des avancées prodigieuses.

L'IA est un outil avant tout venant épauler le travail d'analyses biologiques. A l'avenir, biologie et IA seront amenées à se rapprocher, à converger vers une discipline commune. Les recherches se multiplient pour créer des matériaux biologiques complexes s'approchant des performances humaines.

3. La recherche opérationnelle

La recherche opérationnelle (RO) est au confluent de l'informatique, des mathématiques appliquées, de la gestion et du génie industriel. L'objet de cette discipline est de fournir des bases rationnelles à la prise de décisions, habituellement dans un but de contrôle ou d'optimisation (améliorer l'efficacité, diminuer les coûts, etc.). On utilisera par exemple des techniques de RO pour :

- ✓ Gérer les soins de santé dans les hôpitaux
- ✓ Organiser les services policiers ou ambulanciers
- ✓ Planifier l'utilisation et gérer la production d'énergie
- ✓ Planifier des systèmes de livraison ou de transport en commun
- ✓ Gérer la production, les stocks et la distribution de produits usinés
- ✓ Concevoir des systèmes de communication et des systèmes informatiques
- ✓ Établir des horaires de travail, de cours ou des calendriers sportifs
- ✓ Choisir des politiques économiques et financières

L'objet de cette discipline est de fournir des bases rationnelles à la prise de décisions, habituellement dans un but de contrôle ou d'optimisation (améliorer l'efficacité, diminuer les coûts, etc.) [37].

4. Méthodes d'optimisation combinatoire

Les méthodes de l'optimisation combinatoire peuvent être classées en deux grandes familles de classes : les méthodes exactes et les méthodes approchées. Illustre la taxonomie des méthodes de résolution des problèmes d'optimisation [38].

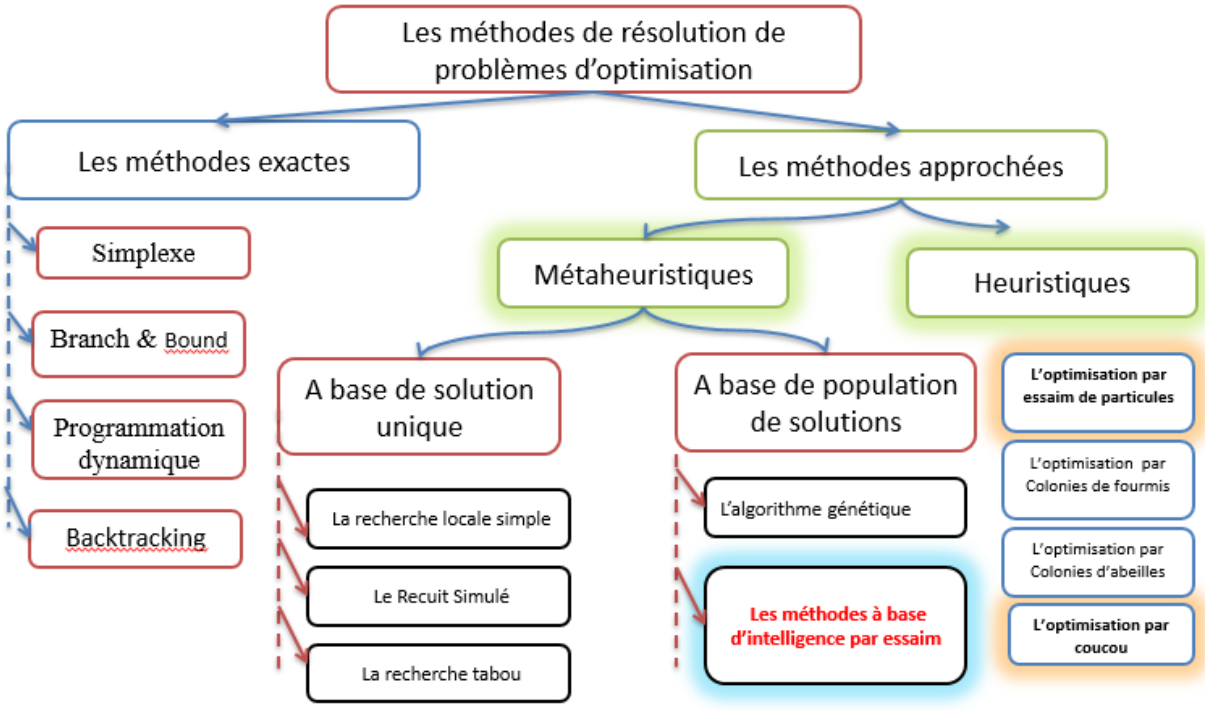


Figure n°08 : Taxonomie des méthodes de résolution de problème d'optimisation. [38]

5. Définition Formelle d'un Alignement Multiple

Soit S = {S1, . . ., SK} un ensemble de séquences définies sur un alphabet Σ. Un alignement multiple d'une matrice d'éléments de $\Sigma \cup \{-\}$ définie par [39] :

$$A=\begin{bmatrix} a_{11}, a_{12} & \cdots & a_{1q} \\ \vdots & \ddots & \vdots \\ a_{k1}, a_{k2} & \cdots & a_{kq} \end{bmatrix}$$

Soit A l'alignement multiple de S1, …, Sk. A est une matrice de dimension K×L avec les propriétés suivantes [41] :

- ✓ $\text{Max}\{n1, \ldots, nk\} \leq L \leq \sum_{i=1}^{K} ni$
- ✓ $A[i][j] \in \Sigma' \quad \forall \quad 1 \leq i \leq K; 1 \leq j \in L$
- ✓ La 1 ère ligne Ai sans gap est égale à Si.
- ✓ Il n'a y a pas de colonnes ne contenant que de gaps.

6. Les méthodes d'alignement multiple des 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 [40].

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 [40].

6.1. L'Approche Exacte

Les algorithmes d'alignement multiples de séquences exacts permettent de réaliser des alignements d'un petit nombre de séquences [41]. L'approche exacte n'est autre qu'une généralisation des méthodes de programmation dynamique de Needleman et Wunsh [34], et Smith et Waterman [42].

La méthode de programmation dynamique utilisée pour aligner deux séquences, a été appliquée à l'alignement de plusieurs séquences (N dimensions) tels que MSA [43] et DCA [44].

Ce type de méthodes représente de gros problèmes : Le temps de calcul et l'espace mémoire [45].

- 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 $2^n L^n$
  - Un espace mémoire proportionnel à L^n

6.1.1. La Programmation dynamique

La programmation dynamique est un principe souvent simple à mettre œuvre pour résoudre des problèmes complexes. Elle ne s'applique toutefois qu`a une certaine catégorie de problèmes, et il est nécessaire de vérifier certaines conditions pour qu'elle puisse être appliquée.

Soit P(n) un problème satisfaisant au principe d'optimalité. On dit qu'un algorithme de résolution de P(n) est basé sur le principe de la programmation dynamique s'il utilise les deux étapes suivantes :

- P(n) est calculé récursivement en partant des problèmes de plus bas niveau.
- Une table est construite dynamiquement pour conserver tous les résultats intermédiaires obtenus [41].

L'utilisation de la programmation dynamique suppose donc que pour les valeurs les plus faibles de n, P(n) soit connu ou facile à calculer. En conservant tous les résultats intermédiaires dans une table on évite d'avoir à les recalculer de nombreuses fois. En effet, refaire ainsi les mêmes calculs à chaque itération est à l'origine de la complexité exponentielle des algorithmes "naïfs" [41].

6.1.2. Méthode basée sur B&B

L'algorithme de B&B (Séparation et évaluation) est une énumération « intelligente » de l'arbre des solutions possibles.

L'algorithme possède deux temps :

1. **La séparation :** séparer un ensemble de solutions en sous-ensembles.
2. **L'évaluation :** évaluer les solutions d'un sous-ensemble en majorant la valeur de la meilleure solution de ce sous-ensemble.

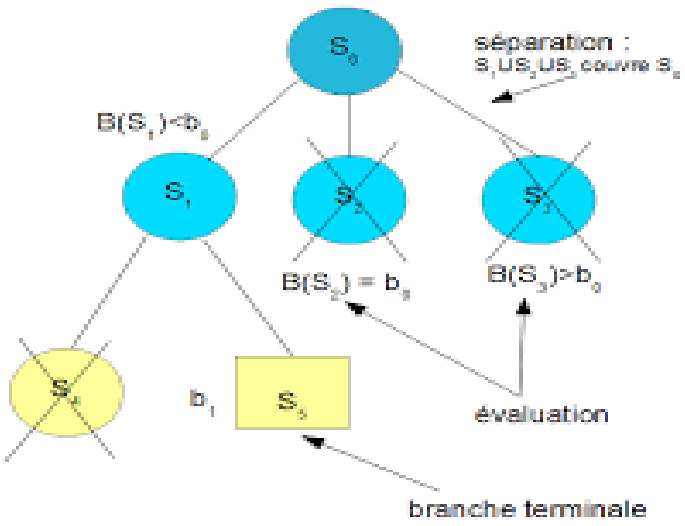


Figure n°09 : La Méthode de B&B [46]

L'algorithme de B&B (Séparation et évaluation) propose de parcourir l'arborescence des solutions possibles en évaluant chaque sous-ensemble de solutions. Il garde en mémoire la valeur de la meilleure solution f(s) trouvée jusqu'à présent. Quand l'évaluation d'un sous-ensemble donne une valeur plus faible que f(s), l'algorithme juge inutile de l'explorer directement après.

L'algorithme de B&B (Séparation et évaluation) est souvent utilisé à la place de la programmation dynamique. L'évaluation est souvent la fonction objective avec des contraintes relaxées (par nombre décroissant) en fonction de la profondeur [46].

6.2. Méthodes Itérative

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 [47].
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.

6.2.1. La Méthode DIALIGN

DIALIGN est une méthode pour l'alignement multiple développée par Morgenstern [48]. 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 [49].
Dialign-t c'est une version plus récente de Dialign-2, locale et progressive.

6.2.2. La Méthode KALIGN

L'algorithme de Kalign suit une stratégie analogue à la méthode progressive standard d'alignement de séquence [50]. Les distances par paire sont calculées, un arbre guide est construit et les séquences/profils sont alignés dans l'ordre donné par l'arbre. Contrairement aux

méthodes existantes, l'algorithme de couplage approximatif de chaînes de Wu-Manber est utilisé dans le calcul de distance et, en option, dans la programmation dynamique utilisée pour aligner les profils [51].

C'est un nouvel algorithme d'alignement de séquences multiples basé sur la correspondance approximative de modèle Wu-Manber qui combine une haute qualité avec une vitesse élevée. Par rapport aux programmes existants, Kalign a réalisé des performances beaucoup plus robustes lors de l'alignement de grandes quantités de séquences ou de séquences distantes dans un repère à grande échelle d'alignements générés. En termes d'efficacité de calcul, Kalign est supérieur aux autres méthodes, et aligne facilement des centaines de séquences en quelques minutes sur un ordinateur de bureau normal. Associé au fait que Kalign donne des alignements très précis, cela fait de Kalign une méthode globale très attrayante. La haute précision de Kalign est due à l'utilisation innovante de l'algorithme de couplage de cordes Wu-Manber approximatif. Cela permet d'estimer avec précision les distances de séquence, même dans les cas difficiles. Des distances de séquence précises génèrent des arbres guides de bonne qualité qui, à leur tour, conduisent à de bons alignements. Dans le même temps, Wu-Manber string-matching est très rapide et réduit considérablement le temps nécessaire pour l'étape d'estimation de distance qui domine le temps d'exécution de la plupart des programmes d'alignement. La stratégie décrite ici peut, en principe, être appliquée à toute autre méthode d'alignement progressif. Même si l'on ne tient pas compte des résultats de la nouvelle grande teste, les performances de Kalign sur Balibase et Prefab sont impressionnantes, surtout si l'on tient compte du fait que, contrairement à d'autres méthodes, Kalign n'a reçu aucune formation sur l'un ou l'autre des tests, et que d'autres méthodes ayant des performances similaires sont beaucoup plus lentes [52].

6.2.3. La Méthode SAGA

C'est un algorithme génétique itératif 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 [53].

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 « seuls les meilleurs survivent ».

Le tableau 2 : Algorithme de SAGA

Initialisation	1. Créer G0
Evaluation	2. Evaluer la population de la génération n (Gn). 3. Si la population est stabilisée, FIN.
Breeding (sélection)	4. Choisir les personnes à remplacer. 5. Evaluer la descendance attendue (OE). 6. Sélectionnez-le ou les parents à partir de Gn. 7. Sélectionner l'opérateur. 8. Générer le nouvel enfant. 9. Conserver ou jeter le nouvel enfant en Gn+1. 10. Passer à 6 jusqu'à ce que tous les enfants aient été placés avec succès en Gn+1. 11. n = n+1 12. Aller à Évaluation.
Fin	13. Fin

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 ont utilisé SAGA pour valider une nouvelle fonction objective : Coffee [54].

Les résultats sont considérés nettement meilleurs que ceux fournis par la première approche [55].

6.3. L'Approche Progressive

L'alignement progressif 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, ClustalW, et T-Coffee …etc. Un alignement multiple progressif suit les étapes suivantes [46] :

a) Alignement deux à deux de toutes les séquences.
b) Construction d'une matrice de distances entre toutes les séquences.
c) Détermination de l'ordre selon lequel les séquences seront Alignées en notion de clustering [44] :
    - ✓ Alignement de deux séquences.
    - ✓ Alignement d'une séquence et d'un profil.
    - ✓ Alignement de deux profils.

6.4. Les Approches basées sur la consistance

6.4.1. La Méthode PCMA

PCMA (**Profile Consistance Multiple Séquence Alignement**) c'un est programme progressif d'alignement multiple des séquences qui combine deux stratégies d'alignement [56].

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 basée sur une nouvelle méthode de comparaison de profil-profil qui est une généralisation de l'approche de PSI-blast 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 [56].

PCMA est une méthode progressive. Elle s'effectue en deux étapes [44] :

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 ClustalW pour constituer un nouveau groupe pré-aligné. À 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 d'E-value sont basés sur une généralisation de l'approche de PSI-blast pour la comparaison profil-séquence [57].

6.4.2. La Méthode PROCONS

PROBCONS est un aligneur multiple basé sur la cohérence.

Les méthodes d'alignement traditionnelles basées sur la distance d'édition évaluent un alignement comme la somme des valeurs de similarité pour les résidus alignés et les pénalités d'écart dépendant de la longueur pour les positions non alignées. Noter un alignement multiple d'une manière probabiliste rigoureuse et biologiquement motivée, et trouver l'alignement optimal une fois qu'un schéma de notation a été spécifié, ne sont pas des tâches simples. En pratique, la mesure ad hoc de la somme des paires, qui combine les distances projetées par paires de toutes les paires de séquences dans l'alignement, est couramment utilisée pour la notation. De nombreuses stratégies heuristiques utilisent un alignement progressif basé sur un arbre évolutif ou des approches itératives, mais sont sujettes à des erreurs dans les premiers stades de l'alignement. Pour lutter contre cela, des étapes de post-traitement telles que le raffinement itératif sont appliquées.

Les systèmes basés sur la cohérence adoptent l'autre point de vue selon lequel "la prévention est le meilleur remède". Pour tout alignement multiple, les alignements par paires induits sont nécessairement cohérents, c'est-à-dire, étant donné un alignement multiple de x, y et z, si la position xi s'aligne sur la position z k et la position z k s'aligne sur y j dans les alignements projetés x-z et z-y respectivement, alors, xi doit s'aligner sur y j dans l'alignement x-y projeté. Les techniques basées sur la cohérence appliquent ce principe en sens inverse, en utilisant des alignements sur des séquences intermédiaires comme preuve pour guider l'alignement par paires de x et y [57].

Ce principe est à la base de l'aligneur PROBCONS.

L'algorithme est décrit ci-dessous [57] :

1. (Alignement initial) Pour chaque paire de séquences x et y.

Calculer un tableau de probabilités postérieur en utilisant le HMM spécifié dans la figure ci-dessous, contenant les probabilités postérieures P (xi ~ yj | x, y) pour faire correspondre chaque lettre xi d'une séquence à chaque lettre yj de l'autre.

Calculer la précision attendue de l'alignement, E (x, y), définie comme étant la somme des probabilités d'appariement postérieures le long du chemin de sommation le plus élevé divisé par la longueur de la séquence la plus courte.

2. (Transformation de cohérence) Mettre à jour simultanément toutes les matrices de probabilité postérieures en utilisant la transformation :

   $P'(x_i - y_j \mid x, y) = \frac{1}{N} \sum_{z \in S} \sum_k P(x_i - z_k \mid x, z) P(z_k - y_j \mid z, y)$

   Répéter cette étape pour un total de deux itérations.

3. (Arbre de guidage) Compte tenu des précisions attendues pour chaque alignement par paire, calculer un arbre de guidage T en utilisant la procédure de regroupement hiérarchique avide suivante
   a) Tout d'abord, placer chaque séquence dans son propre cluster.
   b) Fusionner les grappes x et y avec une précision d'alignement maximale attendue. Lorsque le nouveau cluster x y est formé, définir sa précision attendue avec tout autre cluster z pour être E (x, y) (E(x, z) + E (y, z)) / 2.
   c) Répétez jusqu'à ce qu'il ne reste qu'une seule grappe.
4. (Alignement progressif) Effectuer un alignement multiple progressif selon l'arbre de guidage T en utilisant une fonction objectif somme de paires consistant en la somme des termes $P'(x_i - y_j \mid x, y)$ ré-estimés pour toutes les paires de résidus alignées ; Comme à l'étape 2, aucune pénalité d'insertion n'est utilisée pour calculer le chemin de sommation le plus élevé.
5. (Amélioration itérative) Divisez aléatoirement les séquences dans l'alignement multiple actuel en deux groupes et réalignez. Répétez cette étape pour un total de 100 itérations.

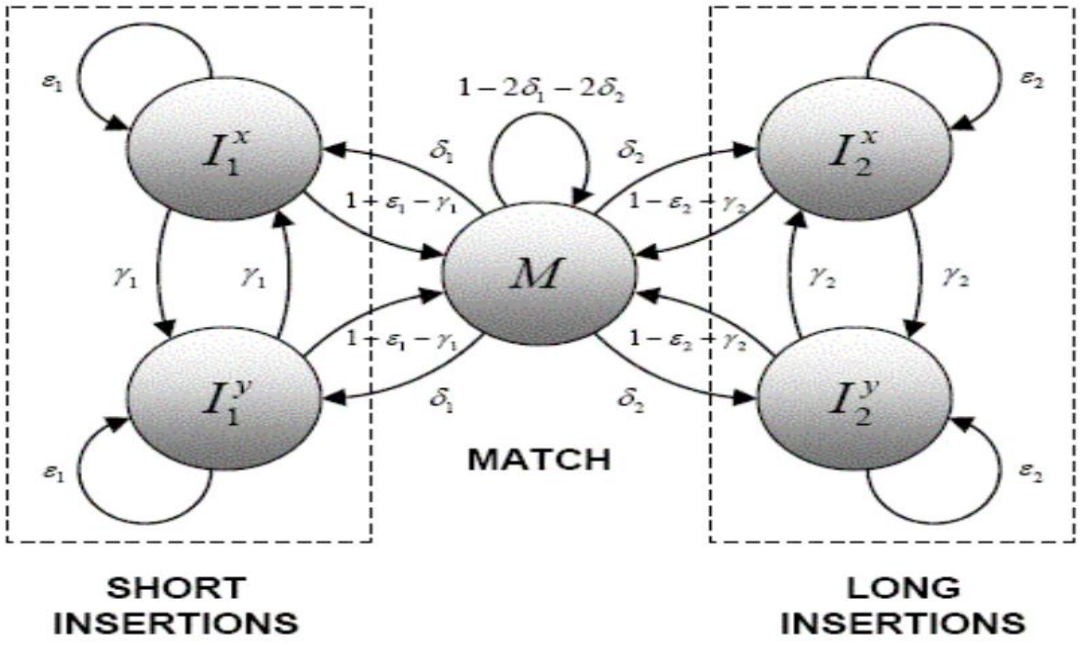


Figure n°10 : Algorithme ProbCons. [67]

6.5. Méthodes évolutionnaires

L'expression « algorithmes évolutionnaires » ou « algorithmes évolutionnistes » (evolutionary algorithms en anglais) désigne une famille d'algorithmes qui s'inspirent de la théorie de l'évolution pour résoudre divers problèmes complexes. Ce domaine a connu son lot de développements au cours des 60 dernières années et il regroupe aujourd'hui une très grande variété d'algorithmes.

Dans le paysage de l'apprentissage automatique, les algorithmes évolutionnaires constituent une technique d'apprentissage non supervisé qui peut être utilisée de façon complémentaire ou alternative aux réseaux de neurones artificiels. Conçue pour résoudre des problèmes d'optimisation, cette technique algorithmique trouve application dans plusieurs domaines notamment pour solutionner certains jeux, pour concevoir des itinéraires, pour développer des voitures autonomes, pour prédire le rendement d'actions cotées en bourses ou encore pour optimiser une chaine d'approvisionnement [58].

6.6. Méthodes d'Alignements Progressives

6.6.1. La Méthode MAFFT

MAFFT 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 [59]

MAFFT a introduit deux 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 [44].

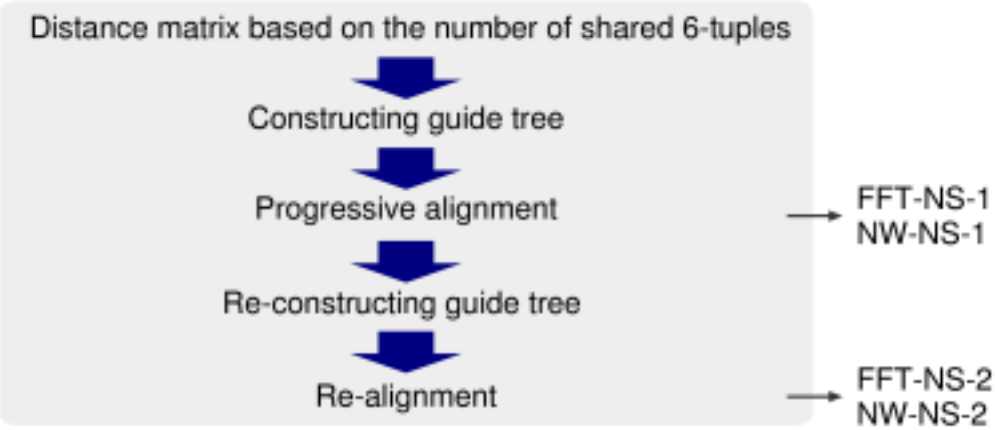


Figure n°11 : les étapes de MAFFT. [44]

6.6.2. La Méthode ClustalW

ClustalW 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 [60].

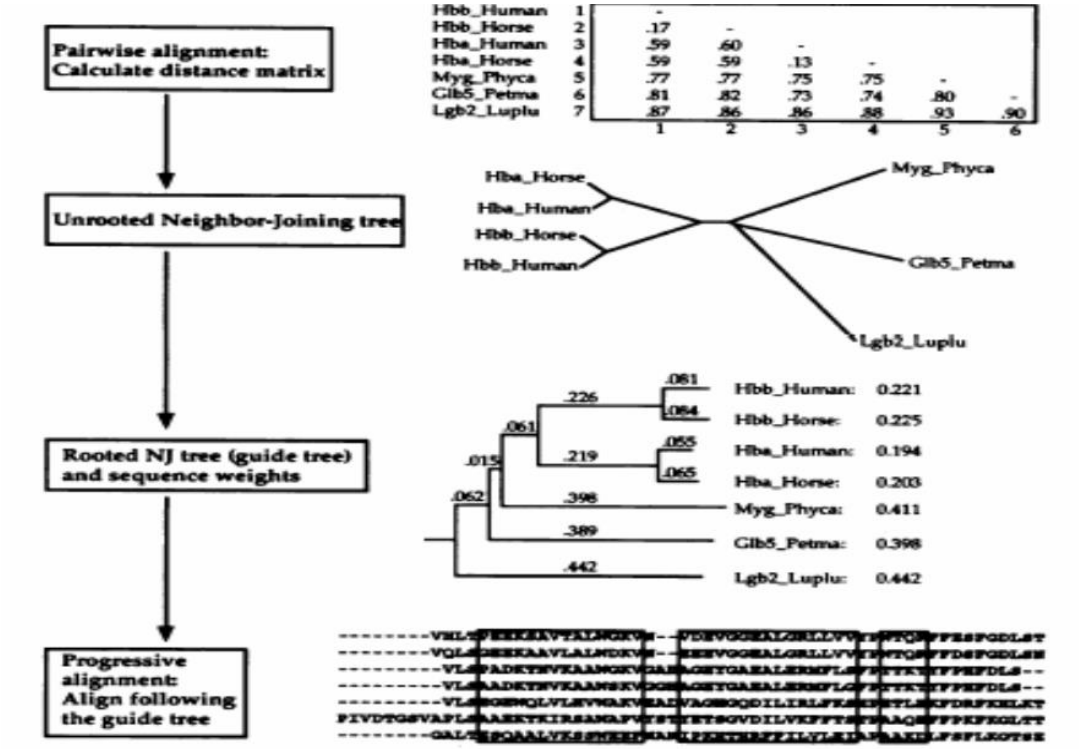


Figure n°12 : Le déroulement de l'algorithme de ClustalW. [44]

- La première étape de ClustalW consiste à aligner les paires de séquences à fin 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 [61].

-Dans la deuxième étape, ClustalW utilise la méthode N.J pour construire un arbre guide et calculer les poids des séquences [61].

-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 [61].

-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 [46].

Une particularité de ClustalW est qu'il possède une interface graphique conviviale contrairement aux autres méthodes [46].

Nouvelle version : Clustal Omega :

- CLUSTAL Omega utilise des arbres guides avec des graines et des techniques de profile-profile HMM pour générer les alignements.
- Changement de plusieurs heuristiques (k-tuple, clustering) de séquence avec les méthodes Bed et k-means, méthode UPGMA pour la construction de l'arbre guide suivie d'un alignement progressif avec le package Hhalign pour produire un l'alignement multiple.
- Clustal Omega est précis et permet l'alignement d'un nombre « infini » de séquences.

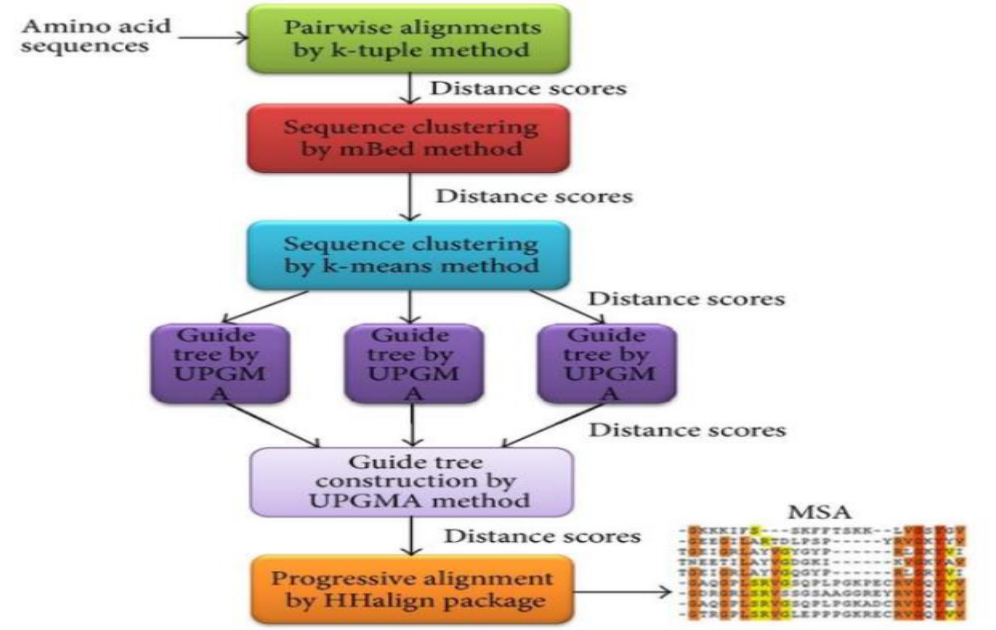


Figure n°13 : Clustal Omega. [44]

6.6.3. La Méthode T-Coffee

T-Coffee (Tree-based Consistency Objective Function for alignement Evaluation) est une méthode qui essaye de pallier les problèmes de l'alignement progressif [62]. Elle fait tout d'abord un prétraitement des données ; construction d'une bibliothèque 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) [62].

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 [62].

Les étapes de base de T-Coffee sont [44] :

1. 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.

2. 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
3. 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.
4. 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.
5. 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.

6.6.4. La Méthode MUSCLE

La méthode MUSCLE 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 [63].

La méthode MUSCLE peut être décrite en trois étapes essentielles [44] :

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 progressif MSA1 est construit alors en suivant l'ordre dicté par l'arbre.

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.

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.

7. Conclusion

Dans ce chapitre ont été introduites les notions de base d'un alignement multiple de séquences, suivies par les principales méthodes d'alignement multiple de séquences publiées et utilisées. Dans le chapitre suivant, nous présentant la méthode sur laquelle nous avons travaillé pour résoudre le problème d'alignement multiple de séquences.

Chapitre 03:

Algorithme d'optimisation par essaim de particules

1. Introduction

L'optimisation par essaims particulaires (PSO : Particle Swarm Optimization) est une des méthodes métaheuristique inspirées de l'intelligence par essaim. Elle s'inspire du mode de vie et d'évolution des essaims pour résoudre des problèmes d'optimisation. Le PSO est un algorithme bio-inspiré. Il repose sur les principes d'auto-organisation qui permettent à un groupe d'organismes vivants d'agir ensemble de manière complexe, à partir de "règles" simples. Le PSO s'inspire du modèle développé par Craig Reynolds pour simuler le déplacement grégaire de certains animaux (troupeaux de bovins, volées d'oiseaux…). Dans ce chapitre, nous allons détailler le principe et les étapes de l'algorithme PSO que nous avons utilisé pour traiter le problème MAS.

2. L'origine de l'idée de l'optimisation par essaim de particules

L'idée de l'optimisation par essaim de particules trouve ses racines dans les années 80. Précisément en 1983, a essayé de résoudre le problème de rendu des images afin de simuler les phénomènes naturels en utilisant l'outil Informatique pour créer des scènes animées. Dans le cadre de son travail, Reeves a implémenté un système de particules qui œuvrent ensemble pour simuler un objet flou (nuage, explosion…). Le modèle proposé par Reeves considère que chaque particule est caractérisée par une position dans l'espace de recherche et une vitesse de déplacement. En effet, les particules de l'essaim se déplacent en fonction de leurs positions courantes et leurs vitesses qui seront adaptées au cours de la recherche [64].

D'autre part, Craig Reynolds a été intrigué par l'organisation et l'esthétique du comportement social des oiseaux. Il a tenté d'améliorer l'idée de Reeves, en rendant le comportement du groupe des particules plus dynamique et plus organisé. Reynolds a ajouté la notion d'orientation et la notion de communication inter-particules : chaque particule doit reste

proche des autres particules de l'essaim comme elle (i.e. la particule) doit éviter d'entrer en collision avec ses congénères. C'est la raison pour laquelle, chaque particule doit avoir conscience de la position des autres particules du groupe. Suite à sa recherche, Reynolds a découvert que l'implémentation d'un modèle simulant le comportement de particules tel qu'il est en réalité n'est pas faisable. En fait, il a découvert que son modèle engendre une exécution très complexe surtout avec une population de grande taille. Afin de pallier ce problème, Reynolds a proposé l'utilisation de la notion du voisinage [65].

3. La PSO de base

Dans un système PSO, un essa1m d'individus parcourt l'espace de recherche. Chaque particule représente une solution candidate au problème d'optimisation. La position d'une particule est influencée par la meilleure position visitée par elle-même (c'est-à-dire sa propre expérience) et la position de la meilleure particule de son voisinage (c'est-à-dire l'expérience des particules voisines). Lorsque le voisinage d'une particule est l'essaim en entier, la meilleure position dans le voisinage est considérée comme la meilleure particule, et l'algorithme est appelé le gbest PSO. Quand des voisinages plus petits sont utilisés, l'algorithme est généralement considéré comme le Lbest PSO. La performance de chaque particule (c'est-à-dire sa proximité de l'optimum global) est mesurée au moyen d'une fonction objective qui varie avec le problème d'optimisation [66].

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 (Figure 14):

- 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 $x_{id}= (x_{i1}, x_{i2}, \ldots, x_{id},\ldots, x_{iD})$ et sa vitesse de déplacement $v_{id}= (v_{i1}, v_{i2},\ldots, v_{id},\ldots, v_{iD})$ 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 : p_{bestid}= (p_{besti1}, p_{besti2},…, p_{bestid},…, p_{bestiD}) et g_{best} = (g_{best1}, g_{best2},…, g_{bestd},…, g_{bestD}).

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 à 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) suivant les deux équations suivantes [67] :

- L'équation (1) :

$$v_{id}(t) = v_{id}(t-1) + c_1 r_1 (p_{bestid}(t-1) - x_{id}(t-1)) + c_2 r_2 (g_{bestd}(t-1) - x_{id}(t-1))$$

- L'équation (2) :

$$x_{id}(t) = x_{id}(t-1) + v_{id}(t)$$

Avec :

- ✓ $x_{id}(t)$, $x_{id}(t-1)$: la position de la particule i dans la dimension d aux temps t et t-1, respectivement.
- ✓ $v_{id}(t)$, $v_{id}(t-1)$: la vitesse de la particule i dans la dimension d aux temps t et t-1, respectivement.
- ✓ $p_{bestid}(t-1)$: la meilleure position obtenue par la particule i dans la dimension d au temps t-1.
- ✓ $g_{bestd}(t-1)$: la meilleure position obtenue par l'essaim dans la dimension d au temps t-1.
- ✓ c_1, c_2 : deux constantes qui représentent les coefficients d'accélération, elles peuvent être non constantes dans certains cas [68] [69].
- ✓ r_1, r_2 : nombres aléatoires tirés de l'intervalle [0,1].
- ✓ $v_{id}(t-1)$, $c_1 r_1 (p_{bestid}(t-1) - x_{id}(t-1))$, $c_2 r_2 (g_{bestd}(t-1) - x_{id}(t-1))$: représentent respectivement, les trois composantes citées au-dessus.

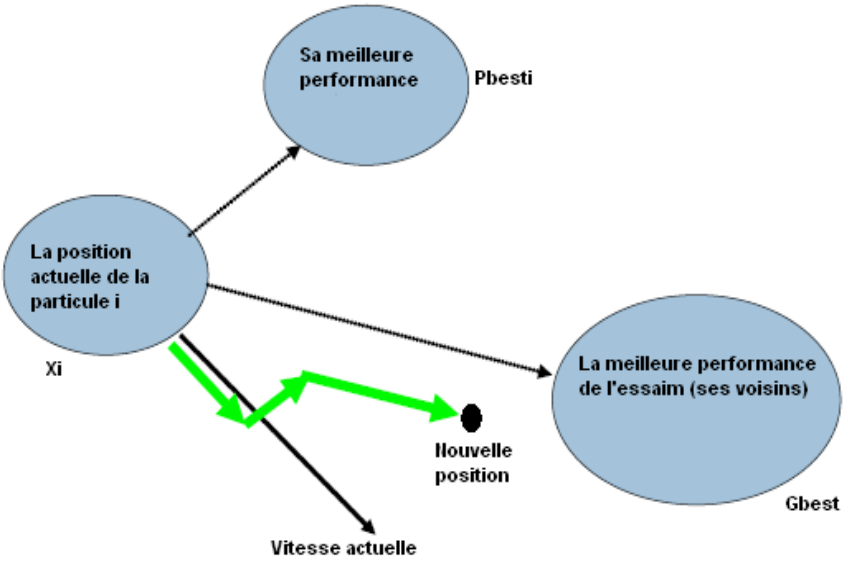


Figure n°14 : Déplacement d'une particule [67]

Afin d'estimer la qualité de la particule i, il est indispensable de calculer sa fonction « objectif » (aussi appelée fitness). La valeur de la fonction « objectif » de la particule xid est notée f(xid). Cette dernière est calculée en utilisant une fonction spéciale au problème traité. Afin de mettre à jour les valeurs de xid, pbestid et gbestd, leurs fitness sont calculés à chaque itération de l'algorithme. xid est mise à jour selon l'équation (2). pbestid et gbestd sont mises à jour si les conditions (C1) et (C2) présentées ci-dessous sont vérifiées respectivement :

F (x_{id}) est meilleur que f (p_{bestid}) (C1)

F (p_{bestid}) est meilleur que f (g_{bestid}) (C2)

L'algorithme PSO commence par initialiser la taille de l'essaim ainsi que les différents paramètres, affecter à chaque particule une position et une vitesse initiales et initialiser les p_{bestid}. Ensuite, calculer les fitness des particules afin de pouvoir calculer la meilleure position trouvée par l'essaim g_{bestd}. À chaque itération du processus de la recherche, les particules se déplacent en fonction d'équations (1) et (2). Leurs fitness sont calculés, les p_{bestid} et la g_{bestd} sont mises à jour. Le processus est répété jusqu'à la satisfaction du critère d'arrêt. L'algorithme indiqué ci-dessous représente un pseudo code de l'algorithme PSO.

Algorithme de L'optimisation par essaim de particules

Début

Initialiser les paramètres et la taille S de l'essaim ;
Initialiser les vitesses et les positions aléatoires des particules dans chaque dimension de l'espace de recherche ;
Pour chaque particule, $p_{bestid} = x_{id}$;
Calculer f (x_{id}) de chaque particule ;
Calculer g_{bestd}; // la meilleure p_{bestid}
Tant que (la condition d'arrêt n'est pas vérifiée) **faire**
Pour (i allant de 1 à S) **faire**
Calculer la nouvelle vitesse à l'aide de l'équation (3.1) ;
Trouver la nouvelle position à l'aide de l'équation (3.2) ;
Calculer f (x_{id}) de chaque particule ;
Si (f (x_{id}) est meilleur que f (p_{bestid})) **alors**
$p_{bestid} = x_{id}$;
Si (f (p_{bestid}) est meilleur que f (g_{bestd})) **alors**
$g_{bestd} = p_{bestid}$;
Fin pour
Fin tant que
Afficher la meilleure solution trouvée g_{bestd} ;

Fin

4. Les variantes de l'algorithme PSO

L'idée des pionniers de l'optimisation par essaim de particules : Kennedy et Eberhart a sollicité l'attention de plusieurs chercheurs qui ont mené des études dans l'objectif d'améliorer la performance de la méthode proposée. En fait, malgré l'efficacité et la simplicité de la mise en œuvre de l'algorithme d'optimisation par essaim de particules, ce dernier souffre du problème de la convergence prématurée [70].

En 1996 Eberhart et ses collègues ont proposé de limiter la vitesse de la particule par l'intervalle $[- v_{max}, v_{max}]$ [71]. Leur objectif était d'échapper au problème de déviation des particules de l'espace de recherche lors de leur déplacement. Le nouveau paramètre v_{max} permet de mieux contrôler le mouvement de particules.

En 1998, Shi et Eberhart ont proposé dans une variante de l'équation (1). La modification apportée consiste à appliquer un facteur d'inertie pour contrôler la vitesse des particules de la manière suivante [72] :

- L'équation (3.3) :

$$v_{id}(t)= \omega\, v_{id}\,(t-1) + c_1\, r_1\,(p_{best}\,(t-1) - x_{id}\,(t-1)) + c_2\, r_2\,(g_{best}\,(t-1) - x_{id}\,(t-1))$$

Où ω est un coefficient d'inertie, généralement une constante qui sert à contrôler l'influence de la vitesse de la particule sur son prochain déplacement afin de garder un équilibre entre l'exploitation et l'exploration de l'espace de recherche. Elle peut être variable dans certains cas: Eberhart et Shi voient que la valeur raisonnable de ω doit diminuer linéairement au cours du processus de l'optimisation. De même Kusum Deep et Jagdish Chand Bansal proposent de réduire la valeur de ω linéairement de 0.8 à 0.4.

De sa part, Clerc a proposé, une autre variante de l'équation (1). Sa variante consiste à ajouter un facteur de constriction K dont l'objectif est de contrôler la vitesse des particules afin d'échapper au problème de la divergence de l'essaim qui cause la convergence prématurée de l'algorithme. L'équation permettant la mise à jour de la vitesse est la suivante [73] :

- $v_{id}\,(t)= K\,[v_{id}\,(t-1) + c_1\, r_1\,(p_{bestid}\,(t-1) - x_{id}\,(t-1)) + c_2\, r_2\,(g_{bestd}\,(t-1) - x_{id}(t-1))]$
- Où $K = \dfrac{2}{\left| 2 - \varphi - \sqrt{\varphi(\varphi-4)} \right|}$

Avec $\varphi = c_{1+} c_{2\ et}\ \varphi > 4$; $c_1 = c_2 = 2.05$; ce qui donne : K=0.729844.

Partant de l'équation (1), une autre variante a été proposée en 2004 par Parsopoulos et Vrahatis Ils ont proposé de calculer la vitesse de la particule à partir d'une combinaison de la vitesse locale et la vitesse globale qui sont définies de la manière suivante [74]:

- ✓ $G(t)= K [v_{id} (t-1) + c_1 r_1 (p_{bestid} (t-1) - x_{id} (t-1)) + c_2 r_2 (g_{bestd}(t-1) - x_{id} (t-1))]$
- ✓ $L(t)= K [v_{id} (t-1) + c_1 r'_1 (p_{bestid} (t-1) - x_{id} (t-1)) + c_2 r'_2(g_{bestld}(t-1) - x_{id} (t-1))]$

  Où G(t) c'est la vitesse globale et L(t) c'est la vitesse locale.

  g_{bestld} c'est la meilleure position trouvée dans un voisinage.

La mise à jour de l'essaim est établie selon les équations suivantes :

- ➢ L'équation (4) : $U(t)= (1-u) L(t) + u G(t)$
- ➢ L'équation (5) : $x(t)= x(t-1) + U(t)$

  U est appelé facteur d'unification, il est tiré de l'intervalle [0,1].

L'équation (4) peut être écrite comme suit [74] :

- ✓ $U(t)= (1-u) L(t) + r_3 u G(t)$
- ✓ Ou $U(t)= r_3 (1-u) L(t) + u G(t)$

  r_3 est un paramètre aléatoire.

Dans un but d'assurer la diversité de l'essaim, Hi ont proposé de leur part de mettre à jour la vitesse des particules selon (l'équation 6) [75] :

- ➢ L'équation (6) :

$v_{id} (t)= \omega\ v_{id} (t-1) + c_1 r_1 (p_{bestid} (t-1) - x_{id} (t-1)) + c_2 r_2 (g_{bestd}(t-1) - x_{id} (t-1) + c_3 r_3 (P^r_{id} (t-1) - x_{id} (t-1)$

Où P^r_{id} est la position de la particule i dans la dimension d de l'espace de recherche. Cette particule est sélectionnée aléatoirement à chaque itération. Le rôle de la composante (P^r_{id} (t-1) - x_{id} (t-1)) est d'assurer la diversité de l'essaim selon la valeur du coefficient d'accélération c_3.

Une autre variante a été proposée par Pongchairerks et Kachitvichyanukul nommée GLNPSO. Afin de mettre à jour la vitesse de chaque particule, Pongchairerks et Kachitvichyanukul se sont basés sur l'équation (3) ainsi que sur la meilleure position locale, globale et celle trouvée dans le plus proche voisinage. L'avantage de l'algorithme proposé (GLNPSO) c'est qu'il permet d'explorer différentes zones de l'espace de recherche

simultanément. La position de la particule est mise à jour selon l'équation (2) alors que la vitesse de la particule est mise à jour selon l'équation suivantes [76] :

- L'équation (7) :

$v_{id}(t) = \omega\,(t)\, v_{id}\,(t\text{-}1) + c_p\, r_1\, (p_{bestid} - x_{id}\,(t\text{-}1)) + c_g\, r_1\, (g_{bestd} - x_{id}\,(t\text{-}1)) + c_l\, r_1\, (L_{bestd} - x_{id}\,(t\text{-}1)) + c_n\, r_1\, (n_{bestd} - x_{id}\,(t\text{-}1))$

Avec : c_p, c_g, c_l et c_n sont des coefficients d'accélération.

L_{bestd}, n_{bestd} sont respectivement, la meilleure position dans un voisinage donné et la meilleure position dans le plus proche voisinage.

Avec $v_{id}\,(t) \in [-\,v_{max}, v_{max}]$ et $x_{id}\,(t) \in [-\,x_{max}, x_{max}]$.

5. Situation de la PSO

Dans la mesure où le concept de 'sawrm' avait été inspiré des automates cellulaires, il serait naturel de voir en quoi il s'en distingue. En effet, au même titre que les particules, les cellules obéissent à des règles simples menant à un phénomène émergent. Au moins trois des propriétés fondamentales que possèdent les automates cellulaires, à savoir : le parallélisme, la proximité (le nouvel état d'une cellule ne dépendent que de son état actuel et de l'état du voisinage immédiat) et de l'homogénéité (les lois sont universelles, c'est-à-dire communes à l'ensemble de l'espace de l'automate), existe aussi pour une population de particules. Il existe pourtant une différence notable au niveau du fonctionnement, les automates cellulaires fonctionnent de manière discrète alors que la PSO a été réfléchie continue (même si des versions discrètes sont venues par la suite).

Une autre analogie s'impose lorsqu'on parle d'approches évolutionnistes. En effet, là aussi il est question de population d'individus évoluant selon des règles simples menant un phénomène émergent. Cependant, la PSO offre une souplesse de formulation alors que le codage dans les algorithmes évolutionnistes les rend rigides.

6. Conclusion

La PSO se caractérise par une cognition d'ordre éthologique donc une forme de cognition sociale très intéressante. En effet, la PSO s'inspire des essaims et mime donc un comportement d'animaux sociaux en vue de résoudre un problème. La PSO a été appliquée avec succès dans

des domaines aussi diversifiés que l'apprentissage de réseaux de neurones, la classification, et l'affectation de tâche.

Dans le chapitre suivant, nous présentons notre méthode, qui consiste à une adaptation du PSO pour résolution du problème MAS.

Chapitre 04 :

Implémentation et discussion

1. Introduction

Dans ce chapitre, nous allons présenter notre contribution qui consiste à utiliser une méta-heuristique pour résoudre le problème MSA. Ce chapitre est organisé comme suit :

Dans une première partie nous allons présenter quelques définitions et notations biologique. Puis nous allons décrire les Méthodes d'alignement multiple des séquences, Où il a mentionné leurs types et classifications, et nous avons fourni une explication simplifiée de chaque méthode et identifié les différences entre elles, puis dans la dernière partie nous avons introduit la méthode sur laquelle nous avons travaillé. Ensuite, nous allons présenter l'implémentation algorithmique de l'approche proposée. Après ça nous allons expliquer le scenario d'évaluation expérimentale ainsi que la présentation et la discussion des résultats. Nous clôturons ce chapitre par une conclusion.

2. La Méthode proposée

Dans notre travail nous avons utilisé le PSO, une méta-heuristique qui a prouvé son efficacité dans plusieurs domaines et qui est considérée parmi les méthodes approchées les plus efficaces. Nous allons présenter un algorithme basé sur PSO appelé PSO-MSA adapté pour le traitement de la problématique suscitée.

Donc l'objectif aussi c'est d'optimiser et d'approcher les chaînes d'acides aminés d'un groupe de protéines à l'aide d'un algorithme PSO.

Cela se fait en créant une application informatique dans le langage de programmation Python, pour que ce travail se fasse automatiquement, rapidement et efficacement ; et l'obtention des résultats se fera dans un dossier contenant toutes les analyses d'optimisation ,et aussi en utilisant un site Web spécial qui mène le processus de l'alignement multiple des séquences biologique.

Description de l'algorithme PSO-MSA :

Début

Initialiser les paramètres et la taille S de l'essaim.
Création de la population des particules
Initialiser les vitesses et les positions aléatoires des particules dans chaque dimension de l'espace de recherche.
Pour chaque particule, $p_{best} = x$
Calculer le score f (x_i) de chaque particule.
Calculer g_{bestd} // la meilleure p_{best}
Tant que (tanque i la condition d'arrêt n'est pas vérifiée) **faire**
Pour (i allant de 1 à S) **faire**

Calculer la nouvelle vitesse à l'aide de l'équation (1)

Trouver la nouvelle position x' à l'aide de l'équation (2)

Calculer le nouveau score f(x') de chaque particule.

Si (f (x') est meilleur que f (p_{best})) **alors**

$p_{bestid} = x'$

Fin Si

Si (f (p_{bestid}) est meilleur que f (g_{best}) **alors**

$g_{best} = p_{bestid}$

Fin Si

Fin pour

Fin tant que

Décoder et afficher la meilleure solution g_{bestd}

Fin

2.1. Création de la population

A partir d'une instance qui contient un ensemble de séquences à aligner, nous cherchons à insérer des gaps tout au long de chaque séquence de l'instance traiter afin d'obtenir un alignement possible, la figure n°15 montre un exemple de création de deux particules à partir d'une instance initiale à aligner.

```
X1{'matrice': [96283, 116517, 178548, 89926, 221617, 96530, 67605, 61527, 8

....N..L.FVALY.D.FV..A..SG..DNT.LSITK...GEK..L.RV.LGY..NHNGE.W.CEAQTKN.G.Q.
..DIDLH..LG..DILT.VN..K..G....S.LV.ALGF...NF...RV.YY..R.DSRDPV....W.KG..P.A



X2{'matrice': [96283, 116517, 178548, 89926, 221617, 96530, 67605, 61527, 8

.NLFV..ALYDF.VAS.GDNT.LSI..T.K..GEKL..RVLG....YN.HNGEW..C..E.AQ..TKN.GQ.GW.'
EE.DI..DL..HLGDI..LTV.N..KGS.LVALG.FS..D..NFRVYYR.DSR.D.PV...W.KG..P..A.K.L
```

Figure n°15 : Les deux particules X1 et X2.

2.2. Prétraitement de donnée

-Nous organisons les séquences des protéines dans des fichiers en Bloc sous format « Txt ».

-Nous allons sur le lien du site web «https://www.ebi.ac.uk/Tools/msa/clustalo/» qui nous fournit le programme «Clustal Omega» pour nous proposer un alignement multiple de séquences de nos instances (voir figure 16).

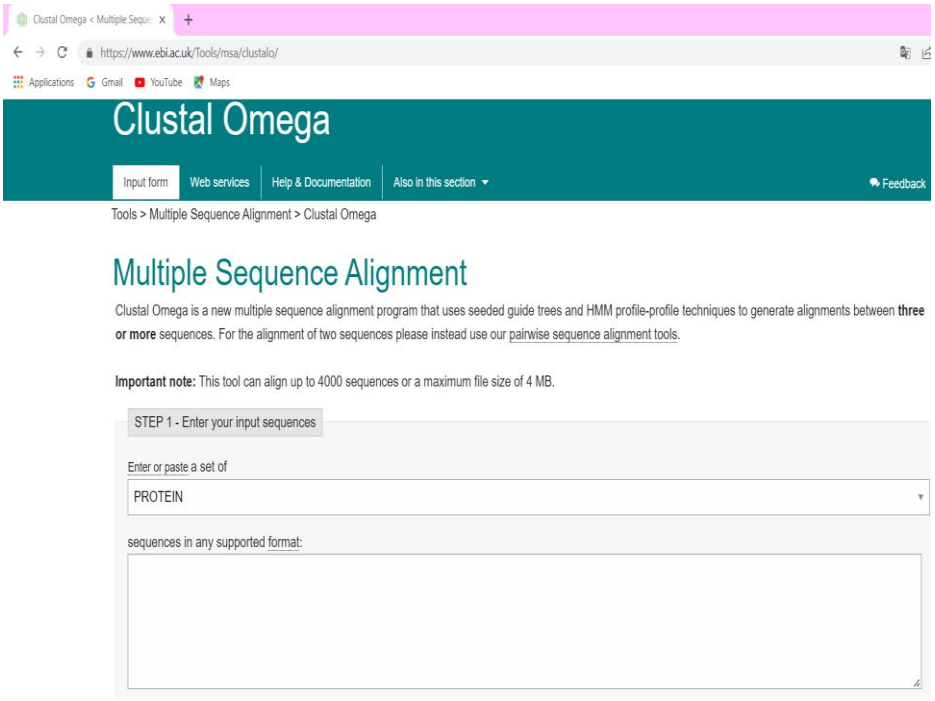


Figure n°16 : l'interface graphique du programme Clustal Oméga.

-Puis nous avons envoyé le fichier au logiciel « Clustal Omega » intégré sur le site web « **www.ebi.ac.uk** », puis appuyez sur « **submit** » pour réaliser l'alignement multiple.

Attendez quelques instants jusqu'à ce que le résultat apparaisse.

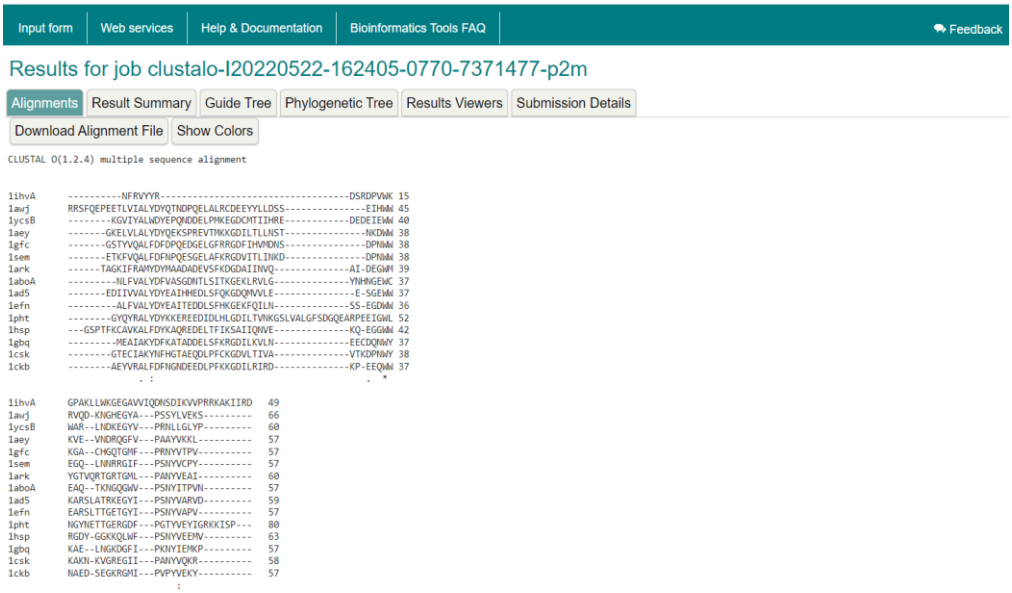


Figure n°17 : Alignement multiple des protéines par la méthode Clustalw.

- Nous avons téléchargé le fichier contenant les résultats de l'alignement multiple، ce fichier porte l'extension «**Fasta**» mais nous allons changer manuellement son extension sur la forme «**Txt**».

-Le fichier que nous téléchargeons et changeons son extension est ce que nous utiliserons dans notre programme.

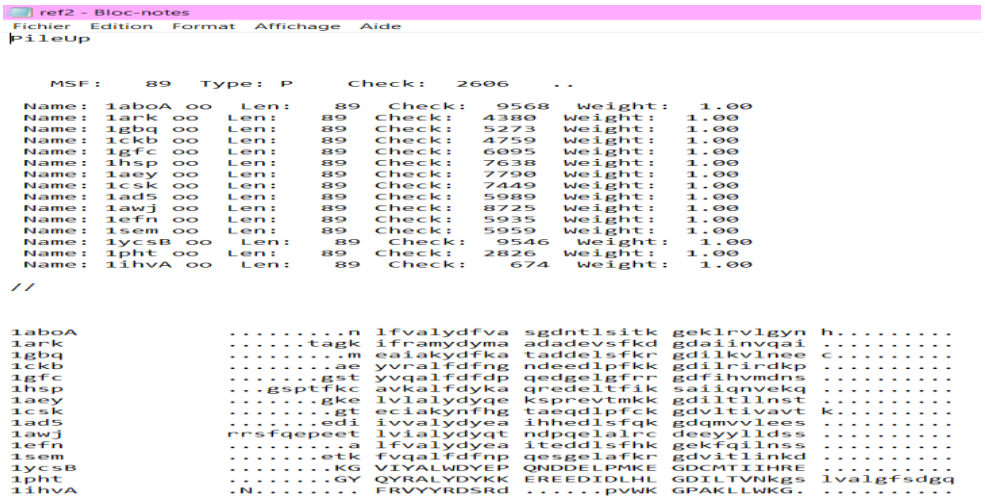


Figure n°18 : Le fichier contenant les résultats de l'alignement multiple

2.3. La représentation des particules

Une particule représente un alignement possible d'un ensemble de séquences protéines, nous avons représentés les différentes particules par des valeurs décimales.

L'exemple suivant montre le codage des particules, où les protéines sont représentées ici par des particules, et à partir de là, le codage se fait par des lettres qui symbolisent les noms de la protéine. Ces caractères sont codés avec des valeurs numériques binaires donc le caractère est représenté par le bit 0 et le Gap (-) par le bit 1, comme illustré dans la figure n°19 ci-dessous

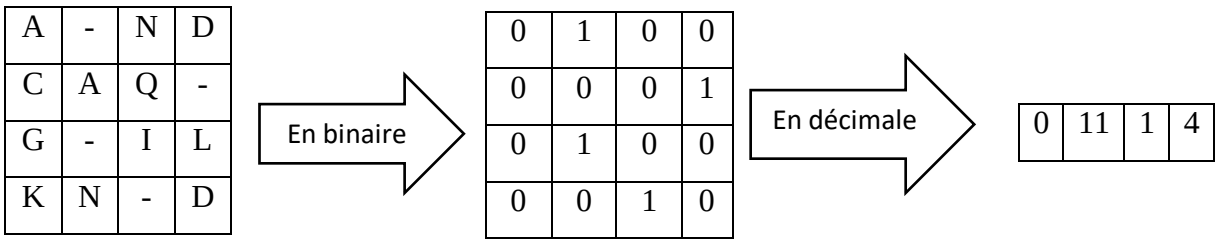


Figure n°19 : Encodage des lettres et les Gaps avec deux chiffres 0 et 1

2.4. Décodage des particules

Nous avons implémenté une fonction qui permet de représenter une particule par des caractères à partir de sa représentation en décimal. Cette fonction est appelée de la mesure de la qualité (la fitness) d'une particule donnée et aussi avant de l'affichage des résultats. La figure n°20 montre le fonctionnement de la fonction de décodage.

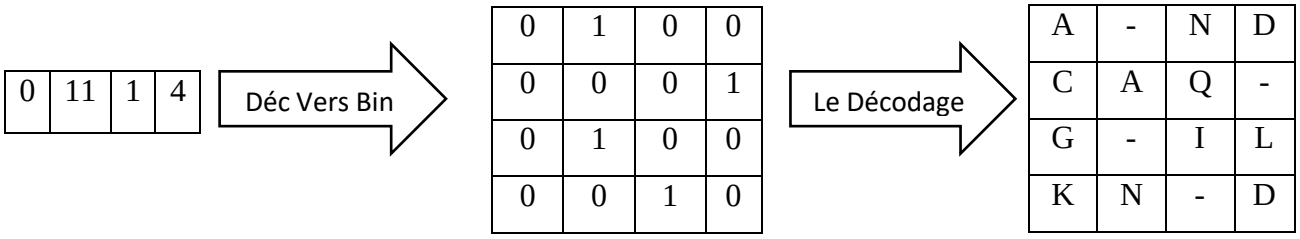


Figure n°20 : Re-décodage de décimal en lettres

2.5. La fonction fitness

C'est une fonction qui permet de calculer le score d'un alignement multiple. Cette fonction utilise les équations suivantes (4.1, 4.2 et 4.3) :

$$S = \sum_{l=1}^{L} S_l \quad (4.1)$$

$$S_l = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} w_{ij} cout(A_i A_j) \quad (4.2)$$

$$W_{ij} = {Nombre\ de\ caractères\ différents\ dans\ l'alignement}/{Longueur\ totale\ de\ l'alignement} \quad (4.3)$$

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 première 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.

3. Data-set utilisés

Nous avons utilisé les instances de la base de données BaliBase 2.0. BAliBASE se compose de 142 alignements de référence, contenant plus de 1000 séquences avec 200.000 résidus. Les alignements sont divisés en dix catégories (La référence 1 ; … ; La référence 10) ; hiérarchiques de référence. Chacune des catégories peut être encore subdivisée en plus petits groupes, selon la longueur de séquence et les pourcentages de similitude [77].

Dans cette contribution, nous avons travaillé sur des instances des références 2 et 3. Les instances utilisées sont :

La référence 2 : (1aboA ;1idy ;1csy ;1r69 ;1tvxA ;1tgxA ;1ubi ;1wit ;2trx ;1sbp ;1havA ;1uky)

La référence 3 : (1idy ; 1r69 ;1ubi ;1uky)

La figure n°21 montre un exemple d'une instance à aligner tirée de la base de données BRALIBASE.

```
>1aboA
NLFVALYDFVASGDNTLSITKGEKLRVLGYNHNGEWCEAQTKNGQGWVPSNYITPVN
>1ark
TAGKIFRAMYDYMAADADEVSFKDGDAIINVQAIDEGWMYGTVQRTGRTGMLPANYVEAI

>1gbq
MEAIAKYDFKATADDELSFKRGDILKVLNEECDQNWYKAELNGKDGFIPKNYIEMKP
>1ckb
AEYVRALFDFNGNDEEDLPFKKGDILRIRDKPEEQWWNAEDSEGKRGMIPVPYVEKY
>1gfc
GSTYVQALFDFDPQEDGELGFRRGDFIHVMDNSDPNWWKGACHGQTGMFPRNYVTPV
>1hsp
GSPTFKCAVKALFDYKAQREDELTFIKSAIIQNVEKQEGGWWRGDYGGKKQLWFPSNYVE
EMV
>1aey
GKELVLALYDYQEKSPREVTMKKGDILTLLNSTNKDWWKVEVNDRQGFVPAAYVKKL
>1csk
GTECIAKYNFHGTAEQDLPFCKGDVLTIVAVTKDPNWYKAKNKVGREGIIPANYVQKR
>1ad5
EDIIVVALYDYEAIHHEDLSFQKGDQMVVLEESGEWWKARSLATRKEGYIPSNYVARVD
>1awj
RRSFQEPEETLVIALYDYQTNDPQELALRCDEEYYLLDSSEIHWWRVQDKNGHEGYAPSS
YLVEKS
```

Figure n°21 : Fichier contient ensemble de protéines de référence 2.

- La première ligne est l'en-tête de chaque séquence et commence par un >: représentant l'identifiant du nom de la protéine.
- La deuxième ligne représente la séquence de la protéine jusqu'à la fin de sa séquence.

- Lorsque la séquence de la première protéine se termine, la séquence commence pour la deuxième protéine jusqu'à la fin de toutes les protéines sur lesquelles nous allons travailler.
- Ces séquences est traditionnellement représenté par une chaîne de caractères qui utilise un alphabet de vingt acides aminés.
- Ces données se présentent en fichier son extension Txt ou Fasta, et contiennent beaucoup d'attributs. Ceux qui ont été utilisées sont : les séquences protéiques.

4. Environnement de travail

- **Python** :

  Est le langage de programmation open source le plus employé par les informaticiens.

  Les principales utilisations de Python par les développeurs sont [78] :

  - ✓ La programmation d'applications
  - ✓ La création de services web
  - ✓ La génération de code
  - ✓ Le méta programmation

- **Anaconda :**

  Est une distribution libre et open source des langages de programmation Python et R particulièrement orientée pour des applications en data science. L'un des atouts majeurs d'Anaconda est sa simplification dans la gestion des packages et de leurs dépendances [79].

- **Jupyter :**

  C'est un éditeur de code pour le langage de programmation Python, permet aux développeurs de partager du code et de l'exécuter dans la même interface utilisateur. Il peut associer du code, des graphiques, des visualisations et du texte dans des notebooks, ou cahiers, partageables qui s'exécutent dans un navigateur [80].

- **Le site web « www.ebi.ac.uk » :**

  C'est une faisons partie du Laboratoire européen de biologie moléculaire (EMBL), une organisation de recherche intergouvernementale financée par plus de 20 États membres, États membres potentiels et associés.

- **Bibliothèques Python utilisées**

  NumPy :

Est une bibliothèque pour le langage de programmation Python qui permet plus de stockage de données avec moins de mémoire. Avec un tableau multidimensionnel et d'autres ressources, Il dispose d'un grand nombre de fonctions mathématiques qui peuvent être appliquées directement à un tableau. Dans ce cas, la fonction est appliquée à chacun des éléments du tableau [81].

Random :

Est un module Python regroupant plusieurs fonctions permettant de travailler avec des valeurs aléatoires. La distribution des nombres aléatoires est réalisée par le générateur de nombres pseudo-aléatoires Mersenne Twister, l'un des générateurs les plus testés et utilisés dans le monde informatique [82].

Biopython :

Le Module Biopython est un projet Gratuit et *Open Source* fournissant des fonctions et des procédures conçues pour le traitement et l'analyse de données biologiques en Python [83].

- L'intérêt principal de ce module est qu'il fournit des parseurs permettant l'accès a de nombreux formats de fichiers utilisés en biologie [82].
- Le module contient des parseurs pouvant lire les enregistrements un par un ou pour les indexer au sein d'un dictionnaire [83].
- Il contient également des modules permettant d'interroger directement par internet des bases de données pour en récupérer les enregistrements (NCBI, ExPASy ...) [83].
- Des interfaces sont développés pour pouvoir utiliser des outils bio-informatiques courants (Blast,Clustal) [83].

La figure n°22 montre un code python permettant l'importation des modules nécéssaire pour l'exécution ne notre programme.

```
1  import os
2  import numpy as np
3
4
5  import random
6  import math
7
8  import pandas as pd
9  import bio
10
```

Figure n°22 : Chargement des bibliothèques.

Le module os :

Est un module fournit par Python dont le but d'interagir avec le système d'exploitation, il permet ainsi de gérer l'arborescence des fichiers, de fournir des informations sur le système d'exploitation processus, variables systèmes, ainsi que de nombreuses fonctionnalités du système [84].

- ✓ **La méthode os.mkdir() :** crée un répertoire correspondant au chemin spécifié.

5. Préparation de l'environnement de travail sur l'ordinateur

- Nous avons organisé deux fichiers en « Bloc-Notes » :

- ✓ Un fichier contenant le résultat de l'alignement multiple des séquences d'un groupe de protéines, nous avons obtenu les résultats d'alignement multiple du programme Clustalw à partir du site Web Clustal Omega.
- ✓ Le deuxième fichier contient la matrice Blosum 62, ce dernier aidez-nous à calculer la somme de l'approche PSO, Il nous permet de choisir le G (best) et le nouveau P (best).

- Créer un nouveau dossier et lui donner un nom 'sortie', et le mettre sur le bureau de l'ordinateur.

- Dans ce dernier dossier, Lors de l'exécution du code que nous avons programmé ; Ce code crée deux fichiers contiennent les résultats de notre méthode.

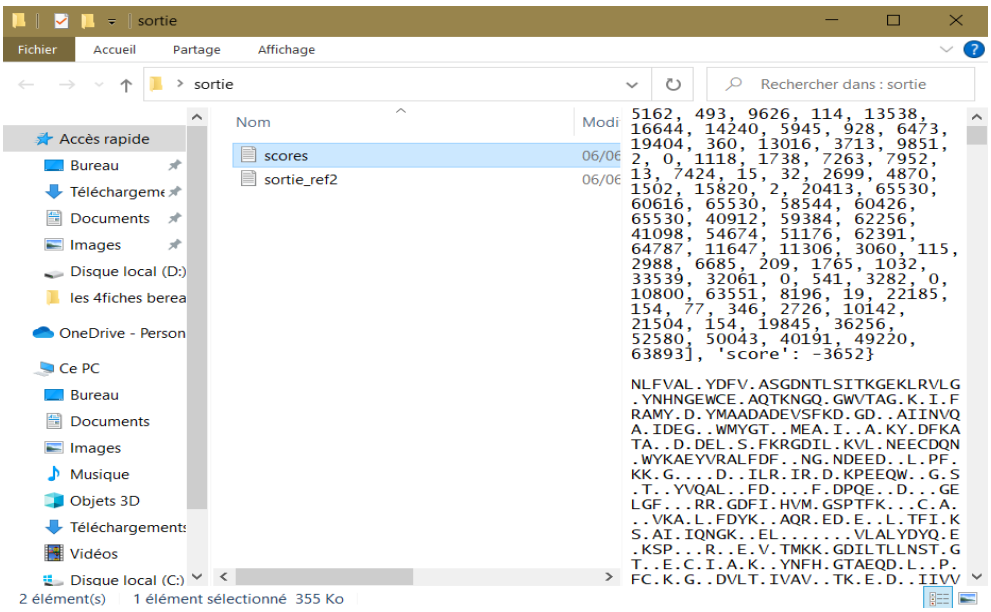


Figure n°23 : Les deux fichiers de résultats d'approche PSO

6. Exécution du programme implémenté

Le fichier est désormais prêt à être utilisé pour classer les données qui ont devenu numérique, Lors de l'exécution du programme en train de regarder le résultat lorsque le mot 'done' apparaît ça veut dire le code est valide.

```
def Score(seq):
    score = 0
    for k in range(len(seq[0])):
        for i in range(len(seq)):
            for j in range(i+1, len(seq)):
                t = (seq[i][k].upper(), seq[j][k].upper())
-------------------------------------------------
-------------------------------------------------
def Mutation(s):
    s = list(s)
-------------------------------------------------.
-------------------------------------------------.
def Function(ar,contenu):
    data_score_m = []
-------------------------------------------------.
-------------------------------------------------.
def binary_to_decimal(binary):
    binary = "".join(reversed(binary))
if "score" not in gbest.keys():
            gbest['score'] = Score(v)
            gbest['ligne'] = "s"+str(i+2)
            gbest['matrice'] = final1
        elif gbest['score'] < Score(v):
            gbest['score'] = Score(v)
            gbest['ligne'] = "s" + str(i+2)
            gbest['matrice'] = final1
    print(gbest)

    print("done")
-------------------------------------------------.
contenu = open("C:/Users/ordinateur/Desktop/ref2.txt").read()
Function("ref2",contenu)
print(cr)
```

Figure n°24 : Les codes les plus importants extraits du programme PSO que nous avons réalisé

```
import numpy as np
blosum = open('./Desktop/blosum.txt').read()
blosum = blosum.split('\n')

top = blosum[0].split(' ')
aside = [x[0] for x in blosum]


top = [x for x in top if len(x)!=0]
aside = [x for x in aside if len(x.strip())!=0]

top[-1] = '.'
aside[-1] = "."

blosum.pop(0)

bl = []
for line in blosum:

    d = line.split(' ')
    d.pop(0)
    d = [ v for v in d if v!=""]

    bl.append(d)

print(bl)
```

Figure n°25 : Le code qui Générer la matrice du Blosum62

7. Récupération des données

On récupère les séquences protéiques sur le bureau de l'ordinateur.

```
contenu = open("C:/Users/Makouf Amir/Desktop/ref2.txt").read()
Function("ref2",contenu)
```

Figure n°26 : Récupération de données.

8. Résultat

Cette présentation suit les étapes du processus et commence par l'exécution du code PSO afin de collecter les lectures et d'extraire si les protéines sont similaires à l'original ou non et prédire la séquence la plus symétrique.

Après l'exécution de notre code de PSO nous avons obtenus ces résultats présentés dans les deux fichiers.

-Le premier fichier contient les séquences de nouvelles particules qui sont proches de la séquence ancestrale.

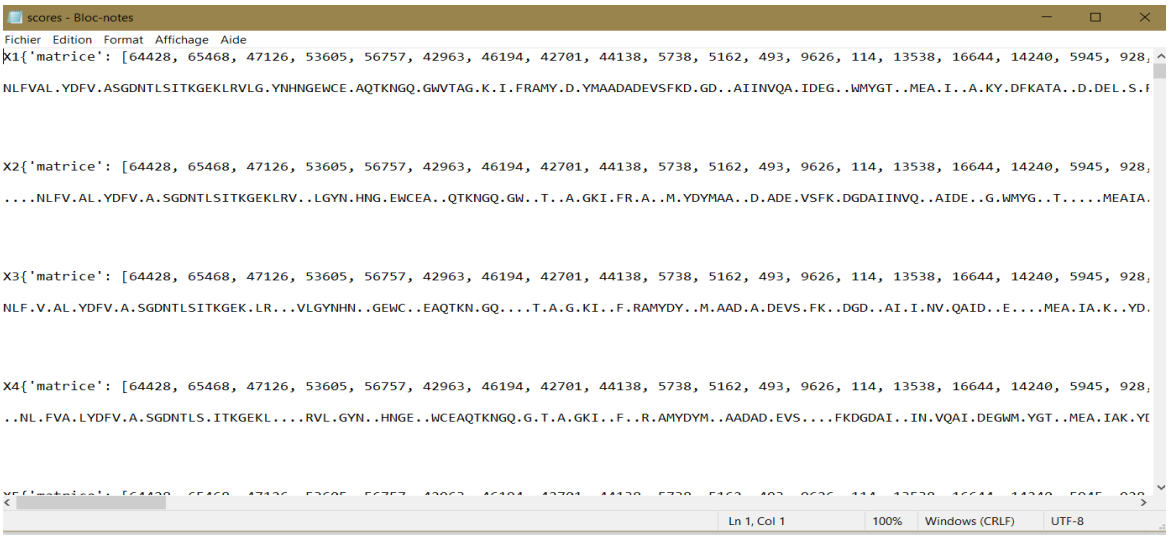


Figure n°27 : Résultat du premier fichier (les séquences de particules) après l'exécution du code de PSO.

- Le deuxième fichier contient le résultat de mutations génétiques survenues au hasard , que nous avons traversé par le système de codage des lettres aux nombres binaires puis des nombres binaires aux nombres décimaux.

```
def Mutation(s):
    s = list(s)
    m = ""
    for i in range(len(s)):
        c = random.randint(0,len(s)-1)
        m += s[c]
        s.pop(c)
    return m
```

Figure n°28 : L'algorithme qui génère des mutations aléatoires

Figure n°29 : Résultat du deuxième fichier (Les nombres décimaux qui codent ces derniers nombres binaires codent la série de mutations).

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-MA, DIALI et PILEUP-8 [85].

Tableau n°03 : Résultats obtenus avec des instances de Réf 2.

Instance	HMMT	ML-PIMA	DIALI	PILEUP-8	PSO-MAS
1aboA	0.724	0.220	0.384	0.000	**0.749**
1idy	0.353	0.000	0.000	0.000	**0.546**
1csy	0.000	0.000	0.000	0.114	**0.780**
1r69	0.000	**0.675**	**0.675**	0.450	0.250
1tvxA	0.276	0.241	0.000	0.345	**0.521**
1tgxA	0.622	0.543	0.630	0.318	**0.780**
1ubi	0.053	0.129	0.000	0.000	**0.740**
1wit	0.641	0.463	0.724	0.476	**0.754**
2trx	0.739	0.702	0.734	0.87	**0.764**
1sbp	0.214	0.054	0.043	0.177	**0.796**
1havA	0.194	0.238	0.000	0.493	**0.790**
1uky	0.395	0.306	0.216	0.562	**0.710**

Tableau n°04 : Résultats obtenu avec des instances de Réf 3.

Instance	HMMT	ML-PIMA	DIALI	PILEUP-8	PSO-MAS
1idy	0.227	0.000	0.000	0.000	**0.524**
1r69	0.000	**0.905**	0.524	0.000	0.410
1ubi	0.366	0.000	0.000	0.268	**0.584**
1uky	0.037	0.148	0.139	0.083	**0.403**

Les deux tableaux 03 et 04 ; représente les résultats obtenus avec les instances de la référence 2 et 3 ; La première colonne représente le nom de l'instance et les autres colonnes représentent les valeurs obtenues par les méthodes : « HMMT ; ML-PIMA ; DIALI ; PILEUP-8 ; PSO-MSA ».

La dernière colonne représente les résultats par la méthode proposée PSO-MSA.

Les meilleures valeurs sont représentées en gras ; notre méthode a obtenu les meilleurs résultats avec 11 instances sur 12 ceci concerne les instances de la référence 2 ; et 3 sur 4 instances concerne les instances de la référence 3.

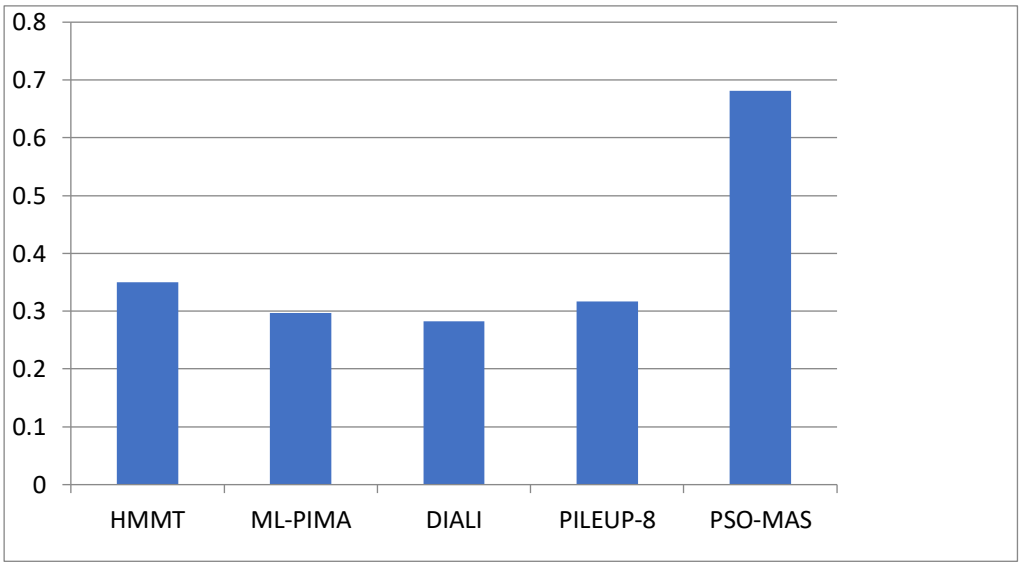


Figure n°30 : La représentation graphique de moy-score des instances de Réf 2.

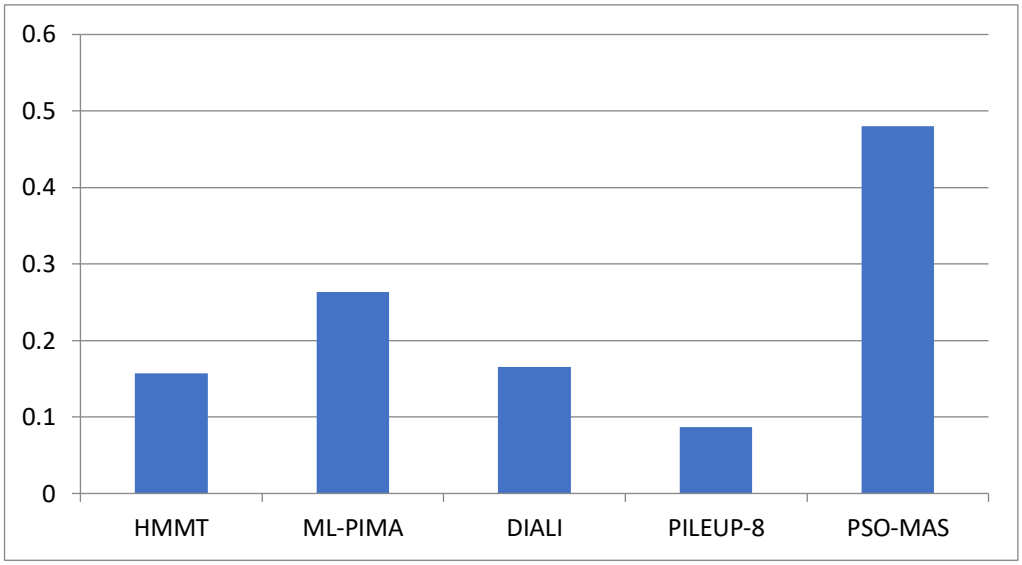


Figure n°31 : La représentation graphique de moy-score des instances de Réf 3.

Les deux graphiques sont présentés dans les figures 30 et 31, représentant respectivement les scores moyens obtenus par chaque méthode avec 12 cas de référence 2 et 4 cas de référence 3.

Comme le montre le graphe ; notre méthode a obtenu les meilleur moyenne des scores.

9. Conclusion

Dans ce chapitre, nous avons présenté une approche basée sur la métaheuristique PSO, l'algorithme effectue une recherche rapide sur un ensemble de séquences protéiques. Nous comparons sa performance avec des algorithmes de recherche exacte basés sur la programmation dynamique dans plusieurs scénarios d'évaluation. Les résultats montrent l'efficacité de l'algorithme proposé.

Conclusion Générale

L'alignement multiple de séquences est une manière de représenter plusieurs séquences de macromolécules biologiques (ADN, ARN ou protéines) les unes sous les autres, de manière à en faire ressortir les régions homologues ou similaires. L'objectif de l'alignement est de disposer les composants (nucléotides ou acides aminés) pour identifier les zones similaires. Cet alignement est réalisé par des programmes informatiques qui visent à détecter et quantifier le nombre de coïncidences entre nucléotides ou acides aminés dans différentes séquences. L'alignement a plusieurs utilisations importantes en bioinformatique car il permet un certain nombre de prédictions. Il permet notamment d'identifier des sites fonctionnels (site catalytique, zone d'interaction...).

Les algorithmes d'alignement multiples de séquences sont très nombreux, ils peuvent prendre des formes très différentes et être basés sur des principes très différents. Il est toutefois possible de les regrouper selon cinq classes :

- L'Approche Exacte
- L'Approche Itérative
- L'Approche Progressive
- Les Approches basées sur la consistance
- Les Méthodes évolutionnaires

Nous avons étudié les aspects théoriques des méthodes d'optimisation. Particulièrement celles basées sur l'intelligence par essaims. Notre étude est basée sur la méthode PSO en tant qu'une des méthodes d'optimisation basées sur l'intelligence par essaims. L'application usuelle de la PSO reste l'optimisation. À partir du moment où la sémantique du problème à résoudre peut s'exprimer sous forme d'une fonction à optimiser, la PSO peut être une bonne méthode pour une résolution efficace.

Notre contribution été la proposition d'un algorithme basé PSO pour l'alignement multiple des séquences. Nous avons proposé une adaptation des procédures de recherche et d'optimisation de l'algorithme PSO 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 de 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.

[1] François Rechenmann,Directeur de recherche Inria, spécialiste de bio-Informatique,03/10/2005.

[3] Katoh, Kazutaka; Misawa, Kazuharu; Kuma, Kei-ichi; Miyata, Takashi (2002). "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform"

[7] Mr BOUZID Allal El Moubarek, Ecole Normale Supérieurs d'Enseignement Technique, ORAN,2008-2009.

[8] Lionel Ranjard, Philippe Cuny, Pierre-Alain Maron, Elisabeth d'Oiron Verame, La Microbiologie moléculaire au service du diagnostic environnemental, ADEME, 2017 (Lire en ligne [archive]), p.13-14

[9] Zhi John Lu, Jason W Gloor, and David H Mathews. Improved RNA secondary structure prediction by maximizing expected pair accuracy. RNA, 15 :1805–1813, 2009.

[10] Ophélie Perrot, Infirmière, membre du comité de rédaction d'infirmiers.com ,11-10-2016

[11] Dr Olivier Perriquet , l'Université des Sciences et Technologies de Lille , le 8-10-2003

[12] Jean-Philippe Vert, Cours de Master Recherche M2 ,2004 /2005

[13] Saïd Berrada, Académie de Dijon, les Protéines : Structure, Propriétés et Applications Technologiques, Journées des 5 et 6 Mai 2009

[16] Olivier PERRIQUET,Doctorat de l'Université des Sciences et Technologies de Lille,le 8-12- 2003.

[17] F. Dardel and F. Képès. Bioinformatique - Génomique et post-génomique. Ellipses, 2002.

[18] Lu, XJ; Bussemaker, HJ; Olson, WK (2 décembre 2015). "DSSR: un outil logiciel intégré pour disséquer la structure spatiale de l'ARN». Recherche sur les acides nucléiques. **43** (21): e142. doi : 10.1093 / nar / gkv716. PMC 4666379. PMID 26184874.

[20] Julien Allali. Comparaison de structures secondaires d'ARN. Informatique. Université de Marne la Vallée, 2004. Français. tel-00637131

[21] B. A. Shapiro and K. Zhang. Comparing multiple RNA secondary structures using tree comparisons. Comput. Appl. Biosci., 6(4) :309–318, 1990.

[22] J. Lamoril ,∗, N. Ameziane , J.-C. Deybach , P. Bouizegarène , M. Bogard , Les techniques de séquençage de l'ADN : une révolution en marche. Première partie DNA A révolution in motion. Part one, Immuno-analyse et biologie spécialisée (2008) 23, 260—279

[23] Gilbert W, Maxam A. The nucleotide sequence of the lac operator. Proc Natl Acad Sci US A1973;70:3581-4.

[24] Sanger F., Nicklen S. Coulson AR DNA sequencing with chain-terminating inhibitors. Proc Natl Acad Sci U S A. 1977;74:5463–5467.

[25] Fleischmann R.D., Adams M.D., White O., Clayton R.A., Kirkness E.F., Kerlavage A.R.Whole genome random sequencing and assembly of Haemophilus influenzae Rd. Science. 1995;269:496–512.

[26] Martin Krahn, Nicolas Lévy et Marc Bartoli , Le séquençage de nouvelle génération monogéniques (Next-Generation Sequencing, ou NGS) appliqué au diagnostic de maladies hétérogènes , publié par EDP Sciences, 2016.

[28] Niranjan Nagarajan et Mihai Pop, « Sequence assembly demystified », Nature Reviews Genetics, vol. 14, 1er mars 2013, p. 157–167

[30] S. B. Needleman et C. D. Wunsch, « A general method applicable to the search for similarities in the amino acid sequence of two proteins », Journal of Molecular Biology, vol.48, no 3, 1er mars 1970, p. 443–453 (ISSN 0022-2836, PMID 5420325, lire en ligne[archive], consulté le 1er avril 2017)

[31] Xia T, SantaLucia J Jr, Burkard ME, Kierzek R, Schroeder SJ, Jiao X, Cox C, Turner DH(octobre 1998). "Paramètres thermodynamiques pour un modèle élargi de plus proche voisin pour la formation de duplex d'ARN avec des paires de bases Watson-Crick". Biochimie. **37** (42): 14719–35. CiteSeerX 10.1.1.579.6653. doi : 10.1021 / bi9809425. PMID 9778347.

[32] Mathews DH, Disney MD, Childs JL, Schroeder SJ, Zuker M, Turner DH (mai 2004). "Incorporant des contraintes de modification chimique dans un algorithme de Programmation dynamique pour la prédiction de la structure secondaire de l'ARN». PNAS. 101 (19): 7287–92. Bibcode: 2004PNAS.101.7287M. Doi: 10.1073 / pnas.0401799101. PMC 409911. PMID 15123812.

[33] Zuker, M. (07/04/1989). "Sur la découverte de tous les replis sous-optimaux d'une molécule d'ARN". La science. **244** (4900): 48–52. Bibcode : 1989Sci ... 244 ... 48Z. doi : 10.1126 / science.2468181. ISSN 0036-8075. PMID 2468181.

[34] Lin, Chien-Ling; Taggart, Allison J. Lim, Kian Huat; Cygan, Kamil J.; Ferraris, Luciana; Creton, Robert; Huang, Yen-Tsung; Fairbrother, William G. (13 novembre 2015). "Lastructure d'ARN remplace le besoin d'U2AF2 dans l'épissage». Recherche sur le génome. **26** (1): 12–23. doi : 10.1101 / gr.181008.114. PMC 4691745.PMID 26566657.

[35] Dr. N. KHERICI, UBMA, 2020/2021

[38] A.Gherboudj. Méthodes de résolution de problèmes d'optimisation difficiles académiques. Thèse de Doctorat. Université Abdelhamid Mehri. Constantine 2. (2013).

[39] Vincent Derrien, Jean-Michel Richer,Jin-Kao Hao." Plasma, un nouvel algorithme progressif pour l'alignement multiple de séquences ". LERIA – Université d'Angers, 2 Bd Lavoisier, 49045 Angers, France.

[40] K. Reinert, "Introduction to multiple Sequence Alignment Algorithmische Bioinformatik, WS, 03, 10, 2003.

[41] Vincent Derrien," Heuristiques pour la résolution du problème d'alignement multiple ",Thése de doctorat, N°d'ordre 885, 2008.

[42] R.C Edgar, "MUSCLE: multiple sequence alignment with high accuracy and high throughput", Nucleic Acids Res. Vol. 32, No. 5, pp. 1792-1797, 2004

[43] T. Smith and M. Waterman, "Identification of common molecular subsequence". *J. Mol. Biol* . Vol. 147, pp. 195-197. 1981.

[44] Nadira Benlahrache, " Optimisation Multi-Objectif Pour l'Alignement Multiple de Séquences ".

[45] J. Stoye, V, Moulton, and A. W. Dress, « DCA, an efficient implementation of the divide approach to simultaneous multiple sequence alignment", Comput. Appl. Biosc., Vol. 13, No. 6, pp. 625-631, 1997.

[46] 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.

[47] Stoye et autres, 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.

[48] 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.

[49] 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.

[50] Feng DF, Doolittle RF: Progressive sequence alignment as a prerequisite to correct phylogenetic trees. J Mol Biol 1987, 25: 351–360.

[51] Wu S, Manber U: Fast Text Searching Allowing Errors. Communications of the ACM 1992, 35: 83–91.

[52] Lassmann, T., Sonnhammer, E.L. Kalign – an accurate and fast multiple sequence alignment algorithm. *BMC Bioinformatics* **6,** 298 (2005).

[53] C. Notredame and D.G. Higgins," SAGA: Sequence alignment by genetic algorithm", Nucleic Acids Res. Vol. 24, No. 8 pp. 1515-1524, 1996.

[54] 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.

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

[56] S.F. Altschul,., T.L Madden, A.A. Schaffer, J. Zhang, Z. Zhang, Z. Miller, and D.J Lipman,"Gapped BLAST and PSIBLAST: a new generation of protein database search programs". Nucleic Acids Res., Vol. 25, pp. 3389–3402, 1997.

[57] Do, C.B., Mahabhashyam, M.S.P., Brudno, M., and Batzoglou, S. 2005. PROBCONS:Probabilistic Consistency-based Multiple Sequence Alignment. Genome Research 15: 330-340.

[58] Simon Du Perron – auxiliaire de recherche au Laboratoire de cyber justice *;* 2020/12/15

[59] Katoh, Kazutaka; Misawa, Kazuharu; Kuma, Kei-ichi; Miyata, Takashi (2002). "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform". Nucleic Acids Research. 30 (14): 3059–66.

[60] 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.

[61] N. Saitou, and M. Nei, "The neighbor-joining method: a new method for reconstructing phylogenetic trees". *Mol. Biol. Evol*., Vol. 4, pp. 406-425. 1987.

[62] C. Notredame, D.G. Higgins and J. Heringa, «T-Coffee: A Novel Method for Fast and Accurate Multiple Sequence Alignment », *J. Mol. Biol*. Vol. 302, pp. 205- 217, 2000.

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

[64] Reeves, 1983

[65] Reynolds, 1987

[66] Nabila Nouaouria, Novembre 2013, UNIVERSITÉ DU QUÉBEC À MONTRÉAL

[67] Kennedy et Eberhart, 1995

[68] Ratnaweera et al, 2004

[69] Kuo et al, 2007

[70] Van den Bergh, 2001

[71] Eberhart et al, 1996

[72] Shi et Eberhart, 1998

[73] Clerc et Kennedy, 2002

[74] Parsopoulos et Vrahatis, 2004

[75] Hi et al, 2004

[76] Pongchairerks et Kachitvichyanukul, 2005

[77] J.D. Thompson, F. Plewniak, and O. Poch, « BAliBASE: a benchmark alignment database For the evaluation of multiple alignment programs". Bioinformatics, Vol. 15, pp. 87– 88, 1999.

[79] Benjamin Berhault, Dec 11, 2018

[80] Serdar Yegulalp, IDG NS (adaptation Maryse Gros), publié le 11 Mars 2019

[85] Taheri, J., Zomaya, A.Y.: RBT-GA: a novel Meta heuristic for solving the multiple Sequence alignment problem. BMC Genom. 10, 1–11 (2009)

Webographie :

[2] https://www.megasoftware.net/web_help_7/hc_clustalw.htm
[4] https://pubmed.ncbi.nlm.nih.gov/12584134/
[5] http://igm.univ-mlv.fr/~dr/XPOSE2013/tleroux_genetic_algorithm/fonctionnement.html
[6] http://acces.ens-lyon.fr/acces/logiciels/applications/geniegen/lalignement-de-sequences-et-Leur-comparaison.
[14] https://www.alloprof.qc.ca/fr/eleves/bv/sciences/la-synthese-des-proteines-s1228
[15] acces.ens-lyon.fr
[19] www-lbit.iro.umontreal.ca
[27] www.clinisciences.com
[29] www.genoscreen.fr
[36] https://lapbm.org/intelligence-artificielle-et-biologie
[37] https://diro.umontreal.ca/departement/quest-ce-que-la-recherche-operationnelle/
[78] www.journaldunet.fr
[81] www.courspython.com/apprendre-numpy.html
[82] www.he-arc.github.io/livre-python/random/index.html
[83] www.irit.fr/~Julien.Pinquier/Docs/TP_MABS/co/Module%20BioPython.htm
[84] www.tresfacile.net/le-module-os-en-python

Année universitaire : 2021-2022

Présenté par : MAKOUF Amir
REBOUH Mounder

Mémoire pour l'obtention du diplôme de Master en Spécialité : Bio-informatique
Domaine : Sciences de la Nature et de la Vie

Résumé

En bio-informatique, l'alignement des séquences est une méthode consistant à représenter deux ou plusieurs séquences de macromolécules biologiques (ADN, ARN ou protéines) pour identifier des régions de similarité qui peuvent être fonctionnelles, structurelles ou évolutives entre les séquences, elles sont considérées comme une partie fondamentale des processus d'une multitude d'applications dans ce domaine qui sert à traiter automatiquement l'information biologique.

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 « optimisation par essaim de particules » (en anglais : Particle Swarm Optimization : PSO). Pour cela, nous avons construit des fonctions pour adapter et utiliser l'algorithme PSO 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-cles : Bio-informatique, Alignements Multiple de Séquence, Méta-heuristique, Algorithme PSO

Encadreur : Dr. Amira GHERBOUDJ ; MCA - Université Frères Mentouri, Constantine 1
Examinateur 1 : Pr. Abdelhafid HAMIDECHI ; Pr - Université Frères Mentouri, Constantine 1
Examinateur 2 : Dr. Hamza CHEHILI ; MCA - Université Frères Mentouri, Constantine 1