الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

N° Réf :.....

Centre Universitaire de Mila

Institut des Sciences et de la Technologie
Département de Mathématiques et Informatique

Mémoire préparé En vue de l'obtention du diplôme de Master

en : - Filière : Informatique

-Spécialité : STIC

- Option

Classification en bioinformatique par des approches basées sur l'intelligence computationnelle

Préparé par :

Djenhi Wiame

Benzebouchi Imene

Soutenue devant le jury :

- Président :	Mme Bouchemal Nardjess	**Grade : MAA**
- Examinateur :	Mlle Bouchekouf Asma.	**Grade : MAB**
- Promoteur :	Mme Afri Faiza.	**Grade : MAB**

Année universitaire : 2013/2014

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

N° Réf :.....

Centre Universitaire de Mila

Institut des Sciences et de la Technologie
Département de Mathématiques et Informatique

Mémoire préparé En vue de l'obtention du diplôme de Master

en : - Filière : Informatique

-Spécialité : STIC

- Option

Classification en bioinformatique par des approches basées sur l'intelligence computationnelle :

Découverte de Motifs par les Algorithmes Génétiques

Préparé par :

Djenhi Wiame

Benzebouchi Imene

Soutenue devant le jury :

- Président :	Mme Bouchemal Nardjess	**Grade : MAA**
- Examinateur :	Mlle Bouchekouf Asma.	**Grade : MAB**
- Promoteur :	Mme Afri Faiza.	**Grade : MAB**

Année universitaire : 2013/2014

Remerciement

Au terme de ce travail, nous saisissons cette occasion pour exprimer nos vifs remerciements à toute personne ayant contribué, de près ou de loin, à la réalisation de ce travail.

Nous souhaitons tout d'abord remercier notre encadreur Mme **AFRI FAIZA** qui nous a encadré avec patience durant la réalisation de ce travail de fin d'études.
Ses conseils nous ont été bien utiles, notamment pour la rédaction de ce mémoire.

L'expression de notre haute reconnaissance à **Mr. Kamel Zeltni** et à Mme **talhi fahima** qui n'ont épargné aucun effort pour se mettre à notre disposition pour la documentation et les explications nécessaires.

Nous exprimons également notre gratitude aux membres du jury, qui nous ont honorés en acceptant de juger ce modeste travail.

Enfin nous tenons à remercier l'ensemble du corps enseignant de la Filière de **Informatique et Biologie**

Merci à Tous

بسم الله الرحمن الرحيم

" *الْحَمْدُ لِلَّهِ الَّذِي هَدَانَا لِهَذَا وَمَا كُنَّا لِنَهْتَدِيَ* لَوْلا أَنْ هَدَانَا اللَّهُ "

الحمد لله أولا و آخرا على فضله وان من علينا لانجاز هذا العمل المتواضع وإن اصبنا فيه فمن الله وان أخطأنا فمن أنفسنا ومن الشيطان

وئام
إيمان

Résumé

La bioinformatique est une discipline qui vise le traitement automatique de l'information biologique. La découverte de motif constitue une tâche fondamentale pour beaucoup d'applications en bioinformatique, est devenu un défit majeur auquel il faut faire face. Les méthodes de découverte de motif proposées dans la littérature reposent sur des approches d'optimisation mono-objectif et fournissent une seule solution potentielle. Le motif joue un rôle important dépendant du type de séquences. Pour les séquences d'ADN, il est lié à l'activité de régulation des gènes. Pour les séquences protéiques, il joue le rôle de signature.

Dans ce travail de master, nous investiguons l'idée de reconsidérer ce problème et de se rapprocher de la solution optimale. La stratégie que nous proposons est l'utilisation des algorithmes génétiques qui a pour but d'extraire le motif commun sous des séquences commune.

Mots clés : Bioinformatique, découverte de motifs, Métaheuristique, les algorithmes génétiques.

Abstract

Bioinformatics is a discipline that aims automatic processing of biological information. The pattern discovery is a fundamental task for many applications in bioinformatics, it has become a major challenge which must be addressed. The pattern discovery methods proposed in the literature are based on mono-objective optimization approaches and provide one possible solution. The pattern plays an important role depending on the type of sequences. In the DNA sequences, it is associated with the activity of gene regulation. For protein sequences, it plays the role of signature.

In this master thesis, we investigate the idea to reconsider this problem and to get closer to the optimal solution. The strategy we propose is the use of genetic algorithms to extract the common pattern under common sequences.

Keywords: Bioinformatics, pattern discovery, métaheuristique, genetic algorithms.

Table des matières

Listes des Figures :

Listes des tables

Introduction générale

Contexte de l'étude :

La Bioinformatique est une science capable de fournir des moyens et des outils pour apaiser la soif des biologistes. C'est un domaine pluridisciplinaire où l'informatique joue un rôle prépondérant. C'est une science qui conceptualise la biologie en termes de molécules et applique des " techniques d'informatiques" pour modéliser, analyser, comparer et simuler l'information biologique incluant séquences, structures, fonctions, …etc. Bref, la bioinformatique est un système intégré de gestion pour la biologie moléculaire et a beaucoup d'applications pratiques.

La découverte de motif est un problème fondamental en biologie moléculaire. Il représente une tâche de base pour beaucoup d'applications en bioinformatique. Il vise à apparier au sens biologique plusieurs séquences nucléiques et protéiques. Cependant, cette extrême importance de la découverte de motif est confrontée à l'extrême difficulté de sa résolution.

Les algorithmes génétiques est un des moyens utilisés par les bioinformaticiens pour analyser des séquences d'ADN (nucléiques) ou de protéines (protéiques) afin de déterminer leur degré d'homologie ou de divergence. Ils sont utilisés pour la découverte des motifs permettent ainsi la prédiction de leur aspect structurel et fonctionnel. Ils adoptent souvent l'exploration d'espaces de recherche très vastes et dont la taille devient de plus en plus critique avec le nombre et les tailles des séquences à étudier.

Durant les dernières années, plusieurs méthodes ont été proposées dans ce domaine. Ce nombre va augmenter les prochaines années du fait que le problème reste toujours ouvert et non complètement résolu.

Le problème de découverte de motifs dans des séquences biologiques est l'un des problèmes les plus connus. La connaissance de ces zones permet de mieux comprendre le fonctionnement des cellules de leur naissance à leur mort, et permet parfois d'expliquer certaines anomalies et certains dysfonctionnements cellulaires. Parmi ces zones, les sites de fixation de facteurs de transcription.

Problématique et motivation :

Avec la grande quantité de données biologiques générée en raison de séquençage d''ADN de divers organismes, il devient nécessaire d'identifier les techniques qui peuvent aider à trouver les informations utiles parmi l'ensemble des données. Trouver des motifs consiste à déterminer des courtes séquences significatives qui peuvent être répétées sur plusieurs séquences de différentes espèces.

Le problème de découverte de ces motifs occupe actuellement une place très importante dans le domaine de la bioinformatique. Il s'agit d'un problème complexe, car on ne sait pas ce que doit être trouvé et les motifs ne sont pas des copies exactes dues aux raisons biologiques.

Malgré que ce problème soit l'un des problèmes classiques d'analyse des séquences biologiques, on n'est toujours pas parvenu à le résoudre de manière satisfaisante, exacte et efficace. Beaucoup d'outils et de méthodes ont été développés par l'évolution des algorithmes de découverte de motif a permis de dynamiser les recherches dans ce domaine. Pour le moment, il n'existe pas d'algorithmes génériques efficaces. Globalement, les métaheuristiques jouent un rôle important dans ce domaine. Elles sont capables de faciliter la mise en place, mais aussi elles donnent des solutions de qualité trouvées en un temps assez raisonnable.

Solution proposée (Objectifs et contributions):

Afin de prendre en compte de manière appropriée la multiplicité des critères de qualité des motifs et la nature combinatoire du problème de leur extraction, nous proposons une solution basée sur une recherche dans un espace mono-objectif via une métaheuristiques pour mettre en évidence des segments d'ADN qui sont susceptibles d'avoir un rôle biologique.

Nous nous intéressons notamment à l'optimisation par les algorithmes évolutionnaires qui constituent un vaste champ de recherche. Ce sont des algorithmes qui concrétisent la politique de l'évolution introduite par Darwin. Les algorithmes évolutionnaires sont donc des algorithmes d'optimisation s'appuyant sur des techniques dérivées de la génétique et de l'évolution naturelle, à savoir: le croisement, la mutation, la sélection, etc.

Par ailleurs, les algorithmes génétiques ont une capacité de résoudre efficacement plusieurs problèmes qui a été démontrée dans plusieurs travaux de découverte de motifs dans les séquences biologiques.

L'objectif de notre travail peut être résumé aux points suivants:

- Etablir une revue des méthodes d'identification de motifs proposées dans la littérature.
- Traiter le problème sous l'angle de l'optimisation mono-objectif et proposer une approche basée sur l'utilisation d'un AG pour résoudre le problème dans le cas de séquences d'ADN et/ou de protéines dans la mesure du possible.
- Implémenter et évaluer l'algorithme.

Structure du mémoire

Cette section vise à présenter l'organisation du mémoire et l'agencement des différents chapitres qui le composent. Le mémoire est divisé en deux parties. La première, appelée état de l'art, vise à introduire les différentes connaissances nécessaires à la compréhension de l'intégralité de ce travail. La seconde partie, nommée apports personnels, présente les résultats des recherches menées durant la durée de travail.

La première partie est composée de trois chapitres. Le premier a pour objectif d'introduire les notions de base de la biologie moléculaire. Le second introduit les concepts de la bioinformatique, ensuite le problème de découverte de motif est exposé, enfin quelques outils liés à la résolution de ce problème sont présentés. Dans le troisième chapitre, le problème d'optimisation est abordé, et l'approche utilisée pour résoudre ce problème est expliquée à savoir les algorithmes génétiques.

La seconde partie est composée d'un seul chapitre, ce dernier présent les outils utilisés pour l'implémentation des algorithmes de la partie précédente, et quelques résultats présentés comme exemples explicatifs.

Nous terminons par une conclusion générale et quelques perspectives.

Chapitre I

Introduction à la Biologie Moléculaire

I. Biologie moléculaire:

1. Introduction :

La biologie moléculaire est apparue au XXe siècle **(le vingtième siècle**), à la suite de l'élaboration des lois de la génétique, de la découverte des chromosomes et de l'identification de l'ADN comme support de l'information génétique.

La biologie moléculaire est essentiellement l'étude des molécules qui constituent les êtres vivants et des processus moléculaires qui assurent leur fonctionnement. Dans cette partie de mémoire nous allons voir une liste non exhaustive de quelques termes clefs et quelques processus moléculaires indispensables pour la compréhension des chapitres suivants.

2. Définition:

Au croisement de la génétique, de la biochimie et de la physique, la **biologie moléculaire** est une **discipline scientifique** dont l'objet est la compréhension des mécanismes de fonctionnement de la cellule au niveau moléculaire.

Le terme « **biologie moléculaire**» désigne également par extension l'ensemble des techniques de manipulations d'acides nucléiques (**ADN**, **ARN**), appelées aussi techniques de génie génétique.

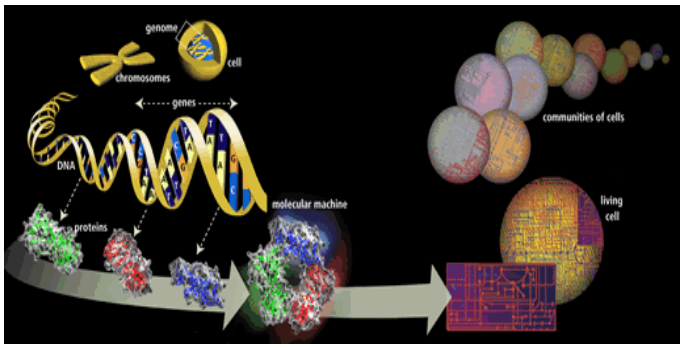

Figure 1 : De la molécule d'ADN à la cellule vivante.

II. Quelques termes-clef de biologie moléculaire et leur définition:

1. La cellule :

La cellule est la brique de structure, fonctionnelle et reproductrice constituant toute partie d'un être vivant. Une cellule est une solution contenant différentes molécules entourée d'une membrane.

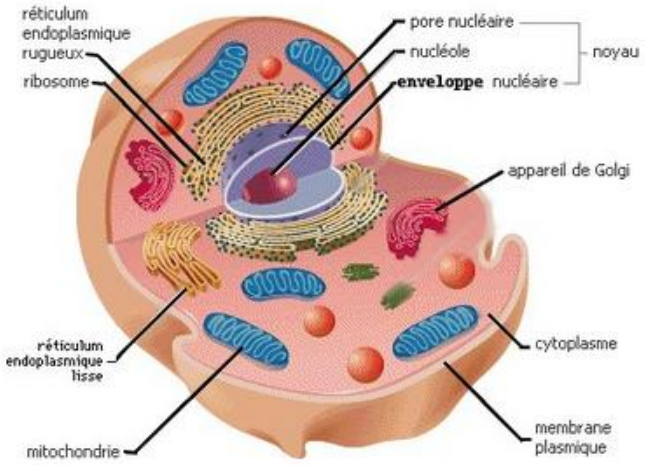

Figure 2 : la structure d'une cellule

Il existe deux types de cellules: Procaryote et Eucaryotes (Figure 3)

- **Les Procaryotes** (du grec **pro**, avant et **karyon**, noyau) sont des êtres unicellulaires, **dépourvus de noyau** et bordés d'une membrane.

- **Les Eucaryotes** vrai noyau en latin délimité par une enveloppe nucléaire isolant du reste leur patrimoine génétique est structuré en chromosomes constitués de plusieurs brins linéaires d'ADN enroulé en double hélice sur des protéines les histones.

Chaque cellule contient la même information mais l'exprime spécifiquement selon sa fonction et son rôle.

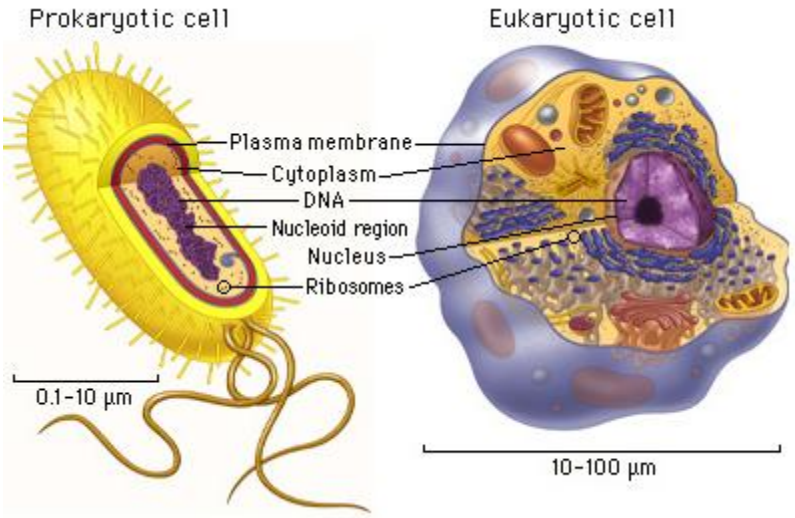

Figure 3 : La structure d'une cellule Eucaryote et procaryote.

- **Eucaryote et procaryote**

L'ensemble des organismes vivants peut être classé en trois grands groupes : les eucaryotes, les eubactéries, les archaebactéries. A l'intérieur de chacune de leurs cellules, les eucaryotes possèdent un noyau : petit sac entouré d'une membrane semi-perméable renfermant les chromosomes. L'Homme, ainsi que les animaux, les plantes et les champignons, sont des eucaryotes. Les eubactéries et les archaebactéries ne possèdent pas de vrai noyau, mais une structure beaucoup plus simple, non entourée d'une membrane. C'est de ce « proto-noyau » que vient le nom générique qui les désigne : procaryotes. Exon / intron / gène mosaïque Chez les eucaryotes, les gènes sont le plus souvent constitués de deux types de séquence nucléotidique : l'une est dite codante et l'autre non codante. Les parties codantes, appelées exons, portent l'information qui sera directement utilisée pour fabriquer les protéines. Entre les exons se trouvent les introns, non « lus » lors de la traduction. Du fait de cette disposition alternée exon/intron, on emploie l'expression gène mosaïque.

Une comparaison très simplifiée est présenté au tableau suivant :

	Procaryotes	Eucaryotes
Organismes	Bactéries, cyanophycées unicellulaire	Protistes, champignons, plantes, animaux, habituellement pluricellulaire
Taille cellulaire	Petite taille 1 – 10 µm de long	Grande taille 10 – 100 µm de long
Information génétique	Nucléöidebactérien, une molécule d'ADN circulaire (+ plasmides)	Noyau entouré d'une enveloppe, ADN + protéines (chromatine)
Type de noyau	nucléoïde (pas de véritable noyau)	vrai noyau avec double membrane
Synthèse des protéines	Polysomes	Polysomes
Gestion de l'énergie	Membrane plasmique	Mitochondries
Division	Scissiparité	Mitose
Cytoplasme	Pas de compartimentintracellulaire Pas de cytosquelette	Compartiments endomembranaires Cytosquelette
Nombre de chromosomes	généralement 1	> 1

Table 1 : Tableau comparatif de l'organisation des Procaryotes et des Eucaryotes.

2. Protéine :

Les protéines représentent 20% du poids de la cellule (eau=70%) L'un des quatre matériaux de base de tout organisme, avec les **glucides**, les **lipides** et les **acides nucléiques**. Les protéines sont formées d'un enchaînement spécifique d'acides aminés (de quelques dizaines à plusieurs centaines).

3. Acide amine (AA)

Petite molécule dont l'enchaînement compose les protéines. On dit qu'une protéine est un **Polymère** d'acides aminés (**les *monomères***).Il existe **20 acides aminés** différents utilisés pour **fabriquer** les **protéines**. **(1)**

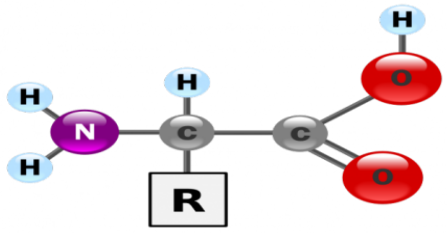

Figure 2 : Acide aminé.

4. Acide nucléique

Polymère formé par l'enchaînement de nucléotides. Les acides nucléiques jouent un rôle fondamental dans le stockage, le maintien et le transfert de l'information génétique. Il existe deux types d'acide nucléique : l'acide ribonucléique (ARN) et l'acide désoxyribonucléique (ADN).

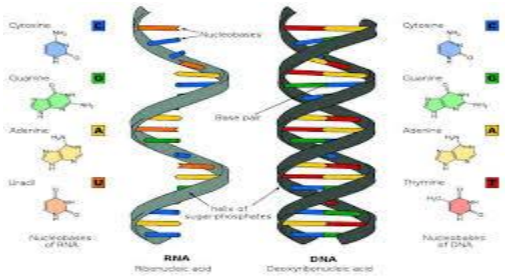

Figure 3 : Acide nucléique.

5. Acide désoxyribonucléique (ADN)

Support biochimique de l'information génétique chez tous les êtres vivants (à l'exception de quelques virus qui utilisent l'ARN) Principal composant des chromosomes, l'ADN se présente le plus souvent sous forme de deux longs filaments (ou chaînes) torsadés l'un dans l'autre pour former une structure en double hélice.

Chacune de ces chaînes est un polymère formé de l'assemblage de quatre nucléotides différents, désignés par l'initiale de la base azotée qui entre dans leur composition : A (Adénine), C (Cytosine), G (Guanine) et T(Thymine).

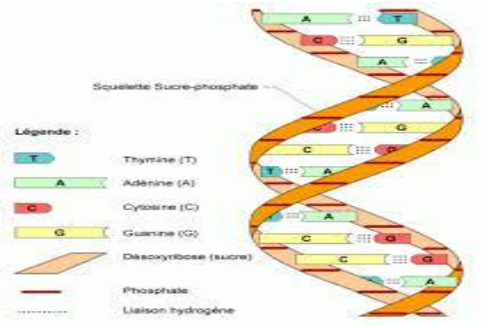

Figure 4: Schéma de la molécule d'ADN.

6. Acide ribonucléique (ARN)

Dans les cellules on distingue plusieurs types d'ARN suivant leur fonction. Les trois types principaux sont : les **ARN messagers**, les **ARN de transfert** et les **ARN ribosomaux**. L'**ARN** est un **acide nucléique** constitué d'une **seule** chaîne de nucléotides, de structure analogue à celle de l'**ADN**. Il existe cependant des différences chimiques entre ces **deux** acides nucléiques qui donnent à l'ARN certaines propriétés particulières. L'ARN est produit par transcription de l'ADN.

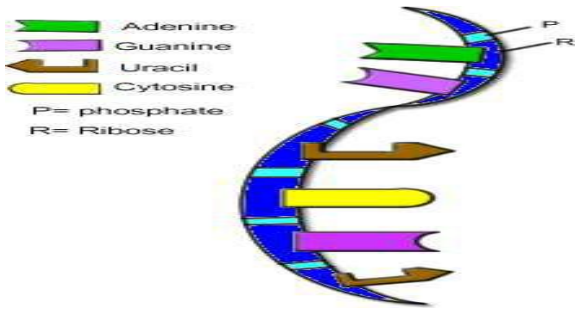

Figure 5: La structure d'ARN.

7. Transcription

La transcription est un processus biologique ubiquitaire qui consiste, au niveau de la cellule, en la copie des régions dites codantes de l'ADN en molécules d'ARN. En effet, si la molécule d'ADN est le support universel de l'information génétique, ce sont les molécules d'ARN qui sont reconnues par la machinerie de traduction en séquences protéiques.

.

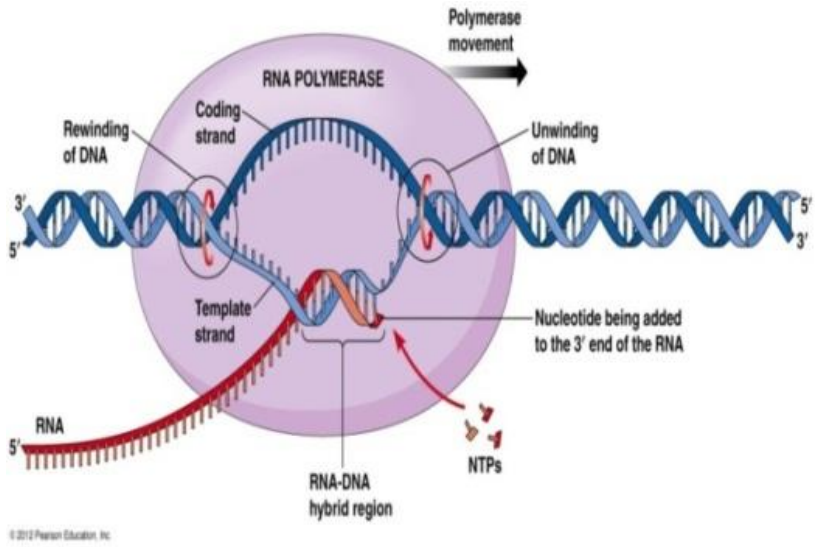

Figure 6:Le mécanisme de la transcription.

L'enzyme qui catalyse cette réaction de transcription est appelée ARN polymérase. Il en existe plusieurs types intervenant dans la transcription de plusieurs types d'ARN (messager, ribosomique, de transfert, etc.) L'ARN polymérase reconnaît et se fixe sur une région particulière de l'ADN, située en amont d'une région codante d'un gène : le site promoteur. Chez les eucaryotes, le transcrit primaire d'ARNm est complété par une queue (polyadénylation) et une extrémité 5 comportant plusieurs modifications chimiques : la coiffe.

La molécule d'ARN directement synthétisée à partir du modèle ADN reste dans le noyau et est traitée par un complexe enzymatique. Ce mécanisme s'appelle l'épissage : certaines séquences appelées introns sont excisées, les exons restant se relient ensuite entre eux. Il peut y avoir un mécanisme d'épissage alternatif,

augmentant ainsi le nombre de possibilités d'ARN messager mature. L'ARN produit est plus court, passe dans le cytoplasme et devient un ARNm ou ARN messager mature.

L'ARNm est alors traduit en protéine à partir des acides aminés en présence des ribosomes et des ARN de transfert (ARNt). Ce mécanisme s'appelle la traduction.

8. Traduction

Étape de la synthèse (fabrication) des protéines au cours de laquelle le brin d'ARN messager obtenu lors de la transcription est converti en une chaîne d'acides aminés qui donnera une protéine. Les ARNm procaryotes sont traduits tels quels, les ARNm eucaryotes subissent une maturation préalable.

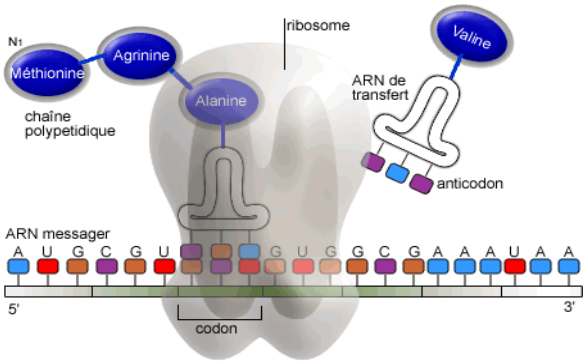

Figure 7 : Le mécanisme de la traduction

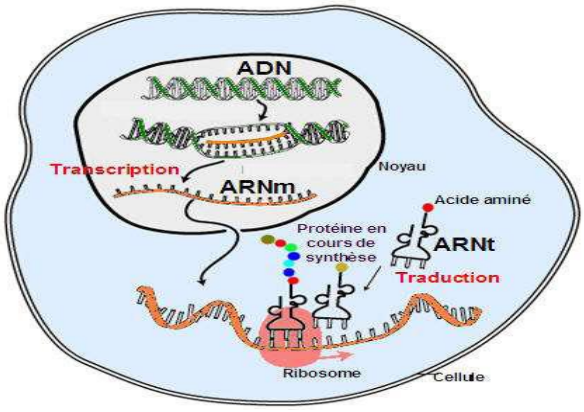

Figure 8 : Récapitulatif ADN ->ARNm-> protéine.

9. Code génétique

Système de correspondance permettant de traduire une séquence d'acide nucléique en protéine. Dans ce système, un triplet de nucléotides ou codon désigne un acide aminé. Comme il existe quatre(4) nucléotides, il y a 4x4x4 = 64 codons différents.

À un codon donné correspond un seul et unique acide aminé. Par contre, il

n'existe que 20 acides aminés différents dans les protéines, c'est pourquoi plusieurs codons peuvent désigner un même acide aminé on dit que le code génétique est **redondant**. Certains de ces 64 codons ne désignent aucun acide aminé. Ces triplets «non-sens» indiquent à la machinerie cellulaire la fin de la lecture de l'information contenue dans les gènes, et provoquent l'arrêt de fabrication des protéines. On les appelle codons STOP.

Tous les êtres vivants (à quelques variantes près) possèdent le même code génétique : il est universel.

Première lettre		*Deuxième lettre*								*Troisième lettre*
		U		C		A		G		
U		UUU	Phénil-alanine	UCU	sérine	UAU	tyrosine	UGU	cystéine	U
		UUC		UCC		UAC		UGC		C
		UUA	leucine	UCA		UAA	codons stop	UGA	codon stop	A
		UUG		UCG		UAG		UGG	tryptophane	G
C		CUU	leucine	CCU	proline	CAU	histidine	CGU	arginine	U
		CUC		CCC		CAC		CGC		C
		CUA		CCA		CAA	glutamine	CGA		A
		CUG		CCG		CAG		CGG		G
A		AUU	isoleucine	ACU	thréonine	AAU	asparagine	AGU	sérine	U
		AUC		ACC		AAC		AGC		C
		AUA		ACA		AAA	lysine	AGA	arginine	A
		AUG	méthionine	ACG		AAG		AGG		G
G		GUU	valine	GCU	alanine	GAU	acide aspartique	GGU	glycine	U
		GUC		GCC		GAC		GGC		C
		GUA		GCA		GAA	acide glutamique	GGA		A
		GUG		GCG		GAG		GGG		G

Figure 9 : Code génétique.

10. Gène

Fragment d'ADN portant les informations nécessaires à la fabrication d'une ou plusieurs protéine(s). Un gène comprend la séquence en nucléotides qui sera transcrite puis traduite en acides aminés, mais aussi des séquences permettant de réguler cette fabrication de protéine en fonction des conditions cellulaires. La longueur d'un gène peut varier de quelques centaines, à plus d'un million de nucléotides.

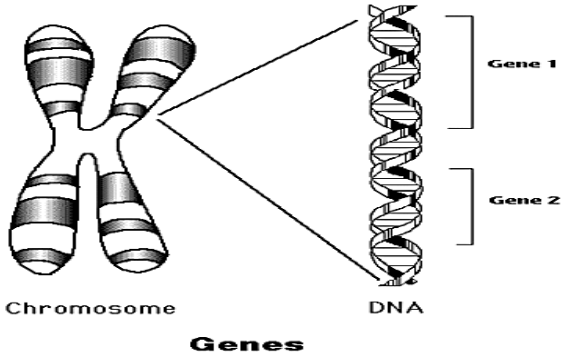

Figure 10: L'emplacement d'un gène dans un chromosome.

11.Génome

Ensemble de l'information génétique d'un **organisme**. Une copie du génome est présente dans chacune de ses cellules. Le génome est transmis de génération en génération. Par extension, le génome se réfère aussi au support physique de cette information génétique, c'est-à-dire la macromolécule d'ADN.

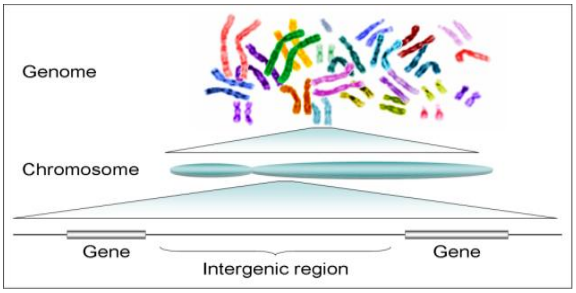

Figure 11: Caryotype.

12.Promoteur

Courte séquence spécifique d'ADN, située au **début** des gènes, sur laquelle se fixe **l'enzyme** qui effectue la **transcription** (l'ARN polymérase). Etant nécessaire pour que la **transcription** débute, le promoteur est **indispensable** au fonctionnement

d'un gène avant de démarrer la synthèse de l'ARN. Les séquences promotrices sont en général situées en amont du site de démarrage de la transcription.

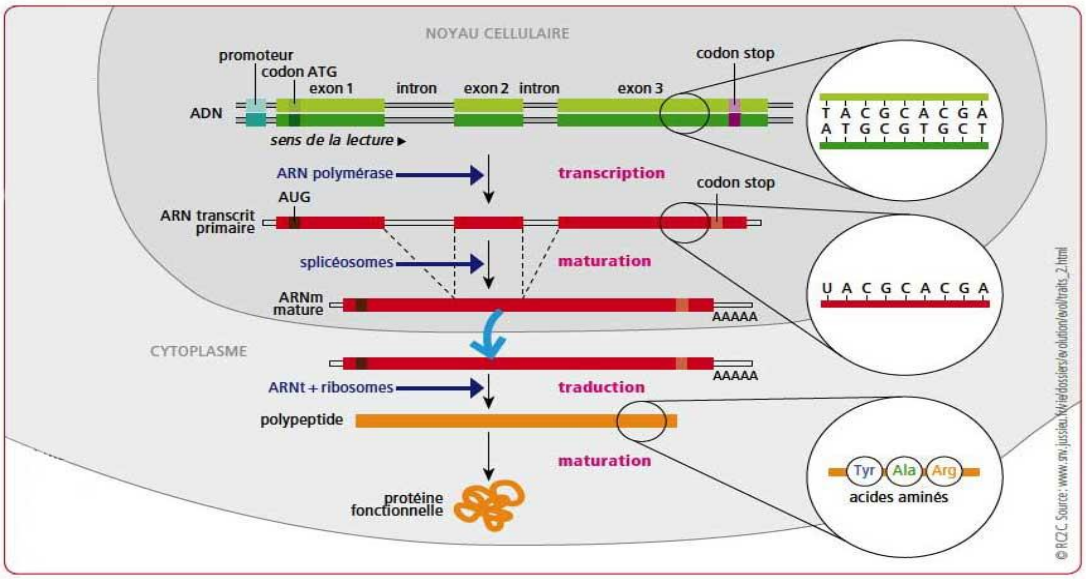

Figure 12 : Organisation générale d'un gène eucaryote.

III. Analyse bioinformatique des séquences :

La mise en évidence de similitude entre séquences sera renforcée si plusieurs séquences voisines issues de plusieurs espèces partagent des éléments en commun. Ceci peut aussi arriver pour une famille de gènes. La méthode permettant d'aligner ces séquences conduit à la mise en évidence des résidus identiques ou similaires conservés pouvant être, pour des protéines par exemple, des éléments clés dans la fonction catalytique ou indispensable à la stabilité d'une structure 3D de la protéine. De même, l'étude de la diversité autour de ces séquences communes, permet par de nombreuses méthodes d'approcher la filiation évolutive de ces gènes et par là même conduire à des études phylogénétiques de plus en plus précises.(2)

o Analyse de séquences :

Une séquence contient les informations sur le rôle biologique d'une macromolécule (ADN/ARN ou protéine): fonction, relation avec les autres molécules .reflète les contraintes physico-chimiques imposées par la fonction l'environnement (aqueux, lipidiques, intra- ou extra- cellulaire) l'évolution moléculaire. Du point de vue d'un bio-informaticien, une séquence biologique est un **MOT**. Un MOT est une **collection ordonnée de symboles** choisis dans un **alphabet** (**A, T, G, C, N**). (3)

Objectif : Prédire des informations pertinentes sur la fonction d'une macromolécule à partir de sa séquence seule.

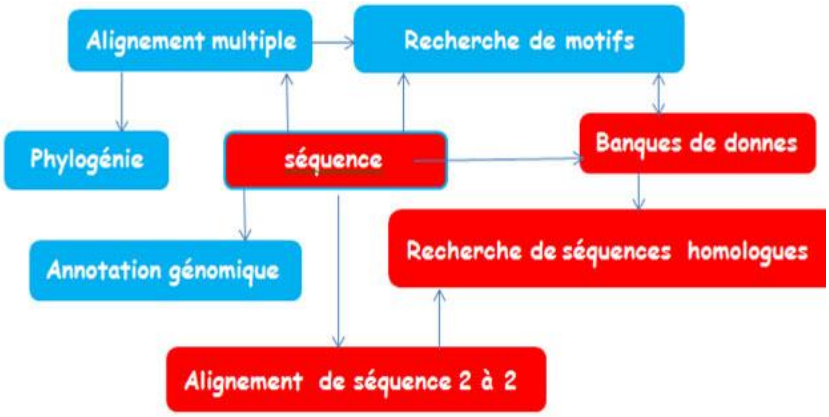

Figure 13 : Domaines d'application d'une séquence.

IV. Conclusion

Dans ce chapitre nous avons présenté quelques éléments de base du vaste domaine de la biologie moléculaire. Suite au projet du génome humain et au développement des technologies sous-jacentes, de grandes masses de données biologiques sont devenues disponibles. Les besoins de leur traitement pour répondre aux divers problèmes non encore résolus à donner naissance à une nouvelle discipline à savoir la bioinformatique. Le chapitre suivant est dédié à l'introduction de ce domaine dont l'un des problèmes posés constitue le contexte de notre travail.

Chapitre II

Bioinformatique Et Découverte de Motifs

I. Introduction :

La **bioinformatique** nouvellement incluse dans les systèmes d'enseignement **biologiques** (elle émerge dans les années **1980**). C'est une **discipline** qui permet **l'analyse** et **l'interprétation** des **informations biologiques** contenues soit dans le génome (séquences **ADN**, **ARN**) soit dans le protéome. On peut également la définir comme étant **la discipline** de **l'analyse** "**insilico**[1]" de **l'information biologique** contenue dans les séquences nucléiques et protéiques.

La révolution extraordinaire que connaît la biologie moléculaire ces dernières années, est en grande partie associée au développement spectaculaire de cette nouvelle discipline qui n'est autre que la bioinformatique. Celle-ci, grâce à ses méthodes qui se basent essentiellement sur les connaissances approfondies de la **biologie moléculaire**, des **mathématiques** et de **l'informatique** (car interdisciplinaire), vient contribuer spectaculairement à l'avancement des connaissances biologiques aussi bien au niveau génomique que protéomique.

On peut considérer que la bioinformatique **tire** sa définition de deux concepts importants : **la biologie** et **l'information** car le suffixe informatique n'a rien à voir avec l'utilisation des ordinateurs pour la biologie. Il s'agit plutôt d'une discipline pour l'interprétation des informations génétiques et structurales.

[1] Insilico les opération qui fait dans la cellule.

II. QUELQUE MOTS SUR LA BIOINFORMATIQUE :

1. Définition de la bioinformatique :

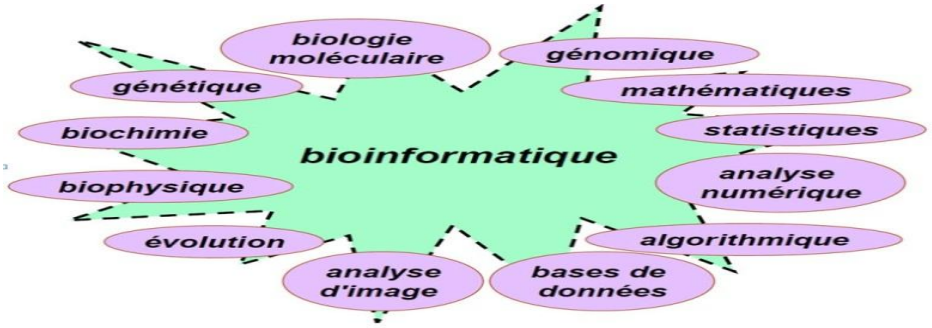

Figure 14 : La bioinformatique et les autres domaines.

La bioinformatique est un domaine de recherche qui analyse et interprète des données biologiques, au moyen de méthodes informatiques, afin de créer de nouvelles connaissances en biologie.

Anglais : distinction entre «Bioinformatics» et « Computational Biology»

Bioinformatics :

- Discipline plus pragmatique.
- Développement d'outils pratiques pour l'analyse et l'organisation des données.
- Moins d'emphase sur l'exactitude ou l'efficacité de la méthode.
- Dédiée à des applications pratiques comme l'identification de protéines cible pour la conception de médicaments.

La biologie computationnelle (Computational Biology) :

- Développement d'algorithmes efficaces permettant de résoudre un problème biologique spécifique.
- Méthodologie générale:
- Définir un modèle d'évolution.

- Formaliser le problème.
- Étudier la complexité théorique du problème.
- Développer des algorithmes permettant de le résoudre.
- S'il y a lieu, prouver l'exactitude de l'algorithme.
- Tester l'efficacité de l'algorithme sur des données simulées.
- L'appliquer à des données biologiques.

Aussi:

Elle est devenue l'outil par excellence pour :

- Interpréter les données biomoléculaires.
- Analyser la structure des molécules.
- Confronter cette structure au reste des molécules existantes dans des bases de données biologiques.
- Prédire le rôle et la fonction de cette structure.

Elle s'intéresse aux données du :

- Génome (totalité du matériel génétique de la cellule).
- Transcriptome (ARNm transcrits).
- Protéome (l'ensemble des protéines bio synthétisées).
- Métabolome (molécules organiques telles que lipides, glucides, faisant partie des activités métaboliques de la cellule vivante).

2. Les applications industrielles de la bioinformatique

La bioinformatique dans son ensemble est donc une activité transverse qui peut être appliquée à de nombreux secteurs des sciences de la vie et des biotechnologies Confrontés à l'utilisation et l'étude du vivant.

Recherche en Biologie

- Organisation moléculaire de la cellule
- Développement
- Mécanismes évolutifs

Médecine

- Diagnostic de cancers
- Détection de gènes impliqués dans le cancer
- Recherche pharmaceutique
- Mécanismes d'action des molécules thérapeutiques
- Identification de cibles thérapeutiques
- Thérapie génique

Biotechnologie

- Bio-ingénierie
- Bio-remédiation (5)

3. Principaux domaines de la bioinformatique

- Bases de données biologiques et outils de requête
- Analyse des séquences nucléiques et protéiques Phylogénie et évolution
- Structures 2D et 3D des macromolécules biologiques
- Analyse des données d'expression génique
- Modélisation et analyse des réseaux biologiques (6)

III. Les banques et les bases de données biologiques :

1. Définition

- **Base de données**

Ensemble structuré et organisé de données permettant le stockage de grandes quantités d'informations afin d'en faciliter l'exploitation (ajout, mise à jour, recherche de données). Evite la redondance. Généralement en champs, organisé en vue de son utilisation par des programmes correspondant à des applications distinctes (gestion, recherche, tri, cartographie, ...). Ce regroupement structuré de données, géré par un système de gestion de base de

données (SGBD), se réalise de manière à faciliter l'évolution indépendante des données et des programmes.

- **Banque de données**

Ensemble pas forcément structuré d'informations, parfois seulement le stockage de références sur des documents. Cousine de la base de données, mais sans contraintes fortes (redondance, cohérence, sécurité, etc.).

Aussi :

Ensemble de données relatif à un domaine défini des connaissances, généralement organisé et structuré en base de données pour être offert aux utilisateurs . On distingue les banques de données bibliographiques (références de documents primaires, avec ou sans résumé), les banques de données iconographiques (images fixes ou animées ; à ne pas confondre avec le mode image), les banques de données textuelles (texte intégral complet ou partiel de documents primaires) ou de type GED (document complet au format original), les banques de données numériques (données chiffrées, plus ou moins structurées) et multimédia (documents construits autour de texte, d'images et de son).

Il existe un grand nombre de bases de données d'intérêt biologique. Nous nous limiterons ici à une présentation des principales banques de données publiques, basées sur la structure primaire des séquences, qui sont largement utilisées dans l'analyse informatique des séquences. Nous distinguerons deux types de banques, celles qui correspondent à une collecte des données la plus exhaustive possible et qui offrent finalement un ensemble plutôt hétérogène d'informations et celles qui correspondent à des données plus homogènes établies autour d'une thématique et qui offrent une valeur ajoutée à partir d'une technique particulière ou d'un intérêt suscité par un groupe d'individus. En biologie, il est fréquent d'appeler les premières "banques de données" et les secondes "bases de données", mais cette distinction n'est pas universelle en dehors du domaine biologique. Aussi, pour éviter toute confusion sémantique

nous parlerons ici de banques de données ou bases de données généralistes (pour les premières) et spécialisées (pour les secondes).

2. Les types des bases de données :

- **Les Bases généralistes**

Les grandes banques de séquences généralistes telles que **Genbank** ou **l'EMBL** sont des projets internationaux et constituent des leaders dans le domaine. Elles sont maintenant devenues indispensables à la communauté scientifique car elles regroupent des données et des résultats essentiels dont certains ne sont plus reproduits dans la littérature scientifique. Leur principale mission est de rendre publiques les séquences qui ont été déterminées, ainsi un des premiers intérêts de ces banques est la masse de séquences qu'elles contiennent. On y trouve également une bibliographie et une expertise biologique directement liées aux séquences traitées. Pour que l'utilisateur puisse s'y repérer, toutes ces informations sont mises à la disposition de la collectivité scientifique selon une organisation en rubriques ou en champs.

- **Les Bases spécialisées :**

Pour des besoins spécifiques liés à l'activité d'un groupe de personnes, ou encore par compilations bibliographiques, de nombreuses bases de données spécifiques ont été créées au sein des laboratoires. Certaines ont continué d'être développées, d'autres n'ont pas été mises à jour et ont disparu car elles correspondaient à un besoin ponctuel. D'autres enfin sont inconnues ou mal connues et attendent qu'on les exploite d'avantage, Toutes ces bases de données spécialisées sont d'intérêt très divers et la masse des données qu'elles représentent peut varier considérablement d'une base à une autre. Elles ont pour but de recenser des familles de séquences autour de caractéristiques biologiques précises comme les signaux de régulation, les promoteurs de gènes, les signatures peptidiques ou les gènes identiques issus d'espèces différentes. Elles

peuvent aussi regrouper des classes spécifiques de séquences comme les vecteurs de clonage, les enzymes de restriction, ou toutes les séquences d'un même génome. En fait très souvent ces bases correspondent à des améliorations ou à des regroupements par rapport aux données issues des bases généralistes. Pour illustrer ce type de banque, nous parlerons ici de bases spécialisées liées aux motifs qui sont particulièrement utilisées dans l'analyse des séquences.

- ## Les bases de motifs

On sait que certains segments d'ADN ou de protéines sont déterminants dans l'analyse des séquences car ils correspondent à des sites précis d'activité biologique comme par exemple les éléments de régulation des gènes ou les signatures peptidiques. C'est pourquoi des bases spécialisées se sont naturellement constituées autour de ces séquences.

- ## Les bases de motifs nucléiques

La plupart de ces bases consiste à recenser dans des catalogues les séquences des différents motifs pour lesquels une activité biologique a été identifiée, Certains motifs sont simples.

Aujourd'hui, il existe principalement deux bases de motifs nucléiques qui sont régulièrement actualisées et qui correspondent à un travail de synthèse bibliographique: il s'agit des bases de facteurs de transcription

- **TFD (Ghosh, 1993)** (6)**: Transcription Factor Database**

TFD est une base dédiée aux facteurs de transcription eucaryotes. Une partie des données a été extraite de GenBank et une autre partie provient de synthèses bibliographiques réalisées à partir de publications traitant de différents aspects de la transcription.

- **TRANSFAC (Knüppel et al, 1994)** (7) (8) :

TRANSFAC est une base de données sur des facteurs de transcription eucaryote et leurs sites de liaisons.

- **Les bases de motifs protéiques**

Il existe principalement deux types de bases de motifs qui permettent de recenser des signatures protéiques liées à des activités biologiques. Celles qui regroupent des motifs consensus et celles qui donnent des régions actives sous forme d'alignements multiples. Nous présenterons ici deux bases couramment utilisées qui reflètent ces deux aspects :

- **la base de motifs protéiques PROSITE**

La base PROSITE peut être considérée comme un dictionnaire qui recense des motifs protéiques ayant une signification biologique.Elle est établie en regroupant, quand cela est possible, les protéines contenues dans Swissprot par famille comme par exemple les kinases ou les protéases. On recherche ensuite, au sein de ces groupes, des motifs consensus susceptibles de les caractériser spécifiquement. La conception de la base repose sur quatre critères essentiels :

- Collecter le plus possible de motifs significatifs.
- Avoir des motifs hautement spécifiques pour caractériser au mieux une famille de protéines.
- Donner une documentation complète sur chacun des motifs répertoriés.
- Faire une révision périodique des motifs pour s'assurer de leur validité par rapport aux dernières expérimentations.

- **La base de motifs protéiques BLOCK**

La base **BLOCK** est également basée sur un système qui détecte et assemble les régions conservées de protéines apparentées. La détection consiste en des alignements multiples à partir desquels des blocs de séquences sont engendrés. Un bloc est la superposition de segments protéiques très similaires sans insertion-délétion. L'ensemble de tous ces blocs forme la base.

C'est ainsi que Henikoff et Henikoff (1991) ont défini 1764 blocs à partir

des 437 groupes de protéines recensés durant l'établissement de PROSITE. Les motifs représentés par la base BLOCK sont généralement plus courts que ceux donnés par la base PROSITE mais les différences fondamentales entre ces bases résident dans la représentation des données. Les motifs de PROSITE sont définis sous forme de chaînes de caractères prenant en compte des insertions et des ambigüités sur les acides aminés conservés alors que les motifs de la base BLOCK sont représentés par des suites d'acides aminés donnés sous forme d'alignements multiples.

L'utilisation de ces bases de motifs est devenue un outil essentiel dans l'analyse des séquences pour tenter de déterminer la fonction de protéines inconnues ou savoir à quelle famille appartient une séquence non encore caractérisée.

3. Exemple de bases de données biologiques :

Nous citerons notamment les bases suivantes qui sont des projets internationaux et constituent des leaders dans le domaine.

EMBL (European Molecular Biology Laboratory) ,GenBank ou encore Swissprot au rang des bases généralistes et la base de structure PDB (Protein Data Bank) et la base Kegg (Kyoto Encyclopedia of Genes and Genomes) comme bases spécialisées parmi les plus connues.

- **EMBL (European Molecular Biology laboratory)[2] :**

Banque européenne de séquences nucléiques créée en 1980 et développée au sein du Laboratoire Européen de Biologie Moléculaire **EMBO (European Moleculary Biology Organisation)** Situé à Heidelberg (Allemagne) :elle est maintenant diffusée par **l'EBI (European Bioinformatics Institute).**

- **GeneBank[3] :**

[2]http://www.ebi.ac.uk/embl/
[3]http://www.ncbi.nlm.nih.gov/genbank/

Une banque de séquences nucléiques créée en 1982 par la société IntelliGenetics et diffusée maintenant par le NCBI (National Center for Biotechnology Information, Los Alamos, US) elle est soutenue par le NIH (National Institute of Health).

- **DDBJ : (Dna Data Base of Japan) une bases de séquences nucléique**

Créée en 1986 et diffusée par le NIG (National Institute of Genetics Japon) Ces trois banques s'échangent systématiquement leur contenu depuis 1987 et ont adopté un système de conventions communes :**"The DDBJ/EMBL/GenBank Feature Table Definition".**

- **PIR-NBRF : Une base de données protéique créée en 1984 par la NBRF (National**

Biomedical Research Foundation). Elle est maintenant un ensemble de données issues Du MIPS (Martinsried Institute for Protein Sequences, Munich, Allemagne) et de la banque japonaise JIPID (Japan International Protein Information Database).

- **Swissprot:**

Une bases de donnée protéique créée en 1986 à l'Université de Genève et maintenue depuis 1987. Celle-ci regroupe aussi des séquences annotées de la banque PIRNBRF ainsi que des séquences codantes, traduites de l'EMBL.

Elle contient 535248 entrées de séquences, comprenant 189901164 acides aminés à partir de 208076 références abstraites. 570 séquences ont été ajoutées depuis la version02/2012, les données de séquence de 127 entrées existantes à été mis à jour et les annotations de 121706 entrées ont été révisées.

Dans plusieurs revues on trouve une distinction entre les bases et les banques

de données.

IV. Différents champs lies à la bioinformatique :

1. la biologie computationnelle

Le développement et l'application des méthodes de données analytiques et théoriques à l'étude de systèmes biologiques, comportementaux et sociaux.

2. Génomique

Étude des ***génomes***. Son objectif est de séquencer l'ADN d'un organisme et de localiser sur celui-ci tous les gènes qu'il porte, puis de caractériser leurs fonctions.

3. La protéomique :

La protéomique est **l'étude du protéome**, dans le but de déterminer l'activité, la fonction et les interactions des protéines, et cela dans diverses conditions. Elle évoque maintenant non seulement toutes les protéines dans une cellule donnée, mais aussi l'ensemble des isoformes de protéines et des modifications, les interactions entre eux, la description structurale des protéines et leurs complexes d'ordre supérieur, et d'ailleurs presque tout.
(8), (9), (10), (11), (12)

4. La pharmacogénomique et la pharmacogénétique :

Les termes «pharmacogénétique» et «pharmacogénomique» sont fréquemment utilisés dans la littérature concernant la médecine personnelle (ou personnalisée). Même s'il n'existe encore aucun véritable consensus sur les

définitions propres à chacun de ces termes, il est néanmoins possible de dégager certaines caractéristiques permettant de les distinguer.

- **La pharmacogénomique :**

Est un terme qui a été introduit à la fin des années 1990 et qu'on peut définir de manière large comme étant l'étude des variations des effets toxiques ou thérapeutiques des médicaments sur la base d'une analyse des informations contenues dans le *génome entier* d'un individu. Les études pharmacogénomique portent aussi bien sur les variations interindividuelles des séquences génétiques elles-mêmes que sur les variations de l'expression des gènes.(13)

Par contraste :

- **La pharmacogénétique :**

Est utilisé depuis les années 50 pour désigner les recherches portant sur des *gènes candidats spécifiques* susceptibles d'expliquer l'existence de variations dans la manière dont un individu répond à un médicament (en termes d'effets secondaires ou d'efficacité clinique, principalement). Les gènes candidats visés dans les études pharmacogénétiques sont sélectionnés en fonction de mécanismes dont on sait déjà ou dont on suspecte qu'ils sont impliqués dans les prédispositions à développer certaines maladies, ou dans l'absorption, le métabolisme, le transport ou l'excrétion de médicaments. Cette sélection de gènes peut également avoir lieu sur la base de cibles médicamenteuses (i.e. des molécules cellulaires ou extracellulaires sur lesquelles se fixent les médicaments et qui jouent un rôle clé dans l'efficacité de ces derniers). Par opposition, la pharmacogénomique suit *une approche libre de toute hypothèse et porte sur l'ensemble du génome.*

5. Pharmaco-informatique:

Pharmaco-informatique se concentre sur les aspects de la bioinformatique portant sur la découverte de médicaments.

6. La génomique structurale ou la bioinformatique structurale :

Se réfèrent à l'analyse de la structure macromoléculaire notamment des protéines, en utilisant des outils informatiques. L'un des objectifs de la génomique structurale est l'extension de l'idée de la génomique, afin d'obtenir une structure précises en trois dimensions des modèles structuraux pour toutes les familles de protéines connues.

7. Génomique fonctionnelle (post-génomique)

Étude de la fonction des *gènes* par analyse de leur séquence et de leurs produits d'expression : les ARNm (transcriptome) et les *protéines* (*protéome*). Elle s'intéresse à leur mode de régulation, et à leurs interactions. L'analyse des protéines peut aller jusqu'à la détermination de leur structure tridimensionnelle.

8. La génomique comparative:

La **génomique comparative** est l'étude comparative de la structure et fonction des génomes de différentes espèces. Elle permet d'identifier et de comprendre les effets de la sélection sur l'organisation et l'évolution des génomes. Ce nouvel axe de recherche bénéficie de l'augmentation du nombre de génomes séquencés et de la puissance des outils informatiques. Une des applications majeures de la génomique comparative est la découverte de gènes et de leurs séquences régulatrices non-codantes basée sur le principe de conservation. (14)

9. Biomédicale informatique / informatique médicale

L'**informatique médicale** est l'application des techniques issues de **l'informatique** au domaine médical. L'informatique médicale est une science à part entière. Aux confluents des sciences de l'information et de la médecine, elle vise à proposer sa contribution pour la compréhension des mécanismes d'interprétation et de raisonnement médical, d'abstraction et d'élaboration des connaissances, de mémorisation et d'apprentissage. La science du traitement de l'information médicale touche aux fondements mêmes de la médecine.

V. Recherche et découverte de motifs:

Avec le volume croissant de séquences biologiques disponible, l'analyse et la découverte de motif est devenu un problème fondamental en biologie moléculaire et représente une tâche de base pour beaucoup d'applications en bioinformatique, ainsi elle est l'un des centres d'intérêt principaux pour un certain nombre de chercheurs de différentes discipline. Il vise à découvrir des modèles significatifs dans les séquences de l'ADN, de l'ARN ou des protéines. L'Intérêt des séquences :

- La séquence nucléotidique d'un gène détermine la séquence d'acides aminés de la protéine.
- La séquence d'une protéine détermine sa structure et sa fonction
- Généralement, une similarité de séquence implique une similarité de structure et de fonction (l'inverse n'est pas toujours vrai)

1. Les motifs biologiques :

Le motif (ou encore signature) peut être défini comme étant une courte séquence (un segment court) continue, non ambigüe et peu dégénérée. C'est une zone fortement conservée le long de l'évolution qui est composée de quelques résidus et est commune à un ensemble de séquences (nucléiques ou protéiques)

ayant la même fonction et le même mécanisme biologiques (séquences homologues). Schématiquement, on peut imaginer le motif comme un petit dessin qui se répète ou non le long d'une séquence donnée. Ce petit dessin sera presque identique chez un bon nombre de séquences ; mais jamais dans les mêmes positions à l'intérieur de ces différentes séquences le contenant.

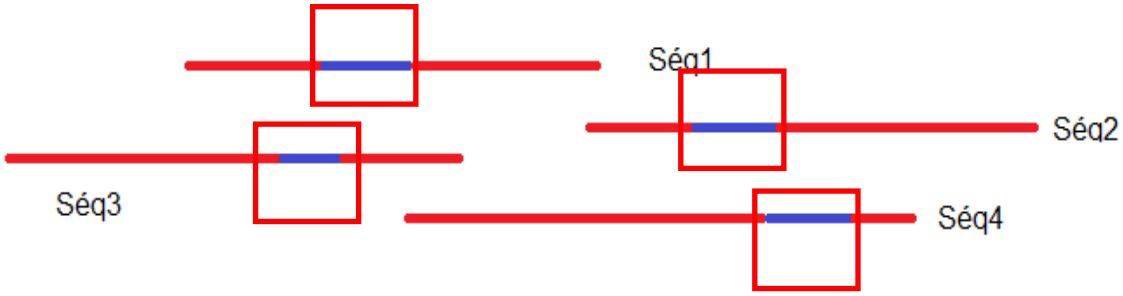

Figure 15 : Le motif dans une séquence.

Le terme **MOTIF** est remplacé par le mot pattern chez les anglo-saxons, sauf que celui-ci peut contenir plusieurs motifs à la fois : Le pattern est une séquence dégénérée et/ou composée de différents motifs avec régions variables. Le motif peut être impliqué dans des fonctions biologiques ou dans des systèmes de régulations.

Comme il peut servir également à identifier une séquence inconnue après confrontation à une base de motifs. (15)

- **Motifs ADN :**

Souvent, ils indiquent des séquences spécifiques des sites de liaison pour des protéines telles que des nucléases, des facteurs de transcription, ou enzyme de restriction. D'autres sont impliqués dans les processus importants au niveau de l'ARN, y compris la liaison au ribosome, maturation de l'ARNm (épissage, polyadénylation) et terminaison de la transcription.

- **Motif protéine :**

Ils peuvent être chargés d'interaction protéine-protéine, le signal de localisation nucléaire (NLS) ou ils peuvent constituer le site actif enzymatique.

Certains motifs sont précis et non ambigus (comme les codons stop ou les sites de coupure d'enzyme de restriction) d'autres peuvent être beaucoup plus flous et complexes (comme les motifs consensus liés à des familles de protéines ou les sites de fixations de facteurs de transcription)[4]. Pour les protéines, les motifs sont généralement étroitement liés à leurs fonctions et structures. Motifs d'ADN sont souvent présents à la région non codante du génome et de servir de signaux pour déterminer les interactions entre l'ADN, l'ARN de transcription, et la machinerie cellulaire. Les motifs sont généralement courts, impliqués dans des systèmes de régulation ou définissent des fonctions biologiques. Il existe donc différentes raisons de les chercher comme par exemple :

- La détermination de la fonction d'une nouvelle séquence
- L'identification de régions codantes dans une séquence nucléique
- La recherche d'un élément transposable dans une séquence
- La recherche d'un site de fixation de facteur de transcription dans une séquence
- La recherche d'un site de coupure d'une enzyme de restriction dans une séquence
- L'extraction de famille de séquences à partir d'une banque de données (exemple: extraire des séquences possédant le même signal de régulation, donc un même motif).

VI. Découverte de Motifs *vs.* Recherche de Motifs

La recherche de ressemblances entre deux ou plusieurs séquences biologiques permet de mettre en évidence certaines fonctionnalités cellulaires. Plusieurs problèmes apparaissent alors, en fonction des informations disponibles. Selon que la recherche de ressemblance est basée sur la connaissance préalable d'un «motif » dans les séquences, ou au contraire qu'il s'agisse de découvrir un ou plusieurs «motifs » ressemblants dans les séquences, il est possible de distinguer deux problématiques complémentaires.

[4]http://genoweb1.irisa.fr/Serveur-GPO/outils/tutoriel/intro_motifs.php

Le premier problème, appelé recherche de motifs dans un ensemble de séquences, consiste à rechercher dans les séquences les positions d'un (ou plusieurs) motif « ressemblants » aux motifs fournis a priori. La notion de ressemblance traduit les phénomènes biologiques d'altération des séquences. Le second problème, appelé « extraction de motifs dans un ensemble de séquences», consiste à extraire le (ou les) motif(s) commun(s) (i.e., qui se ressemblent ou qui ressemblent à un motif consensuel) à l'ensemble de séquences, sans connaissance précise de ce(s) motif(s). Dans les deux cas, la première difficulté rencontrée lors de l'élaboration de méthodes de résolution de ces problèmes, est de définir l'ensemble des critères permettant d'affirmer si deux motifs sont ressemblants ou non. S'en suit généralement le problème de la délimitation de l'espace de recherche, et enfin de l'évaluation de la pertinence des résultats fournis par lesdites méthodes.

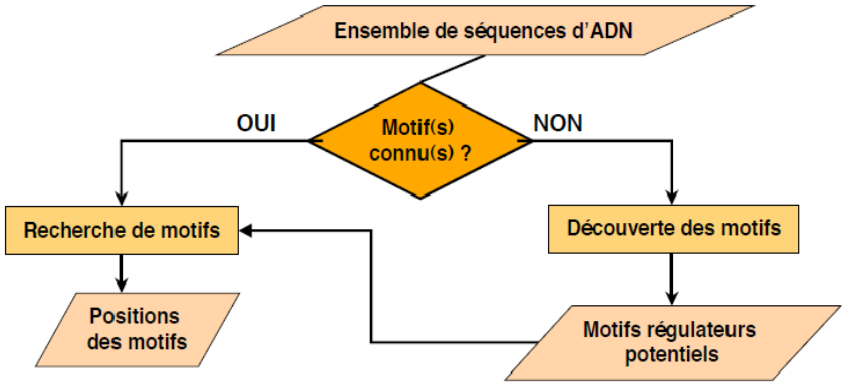

Figure 16 : Découverte et Recherche de motif.

1. La Recherche d'un «Motif » dans une Séquence :

Un "motif" (ou ≪ pattern ≫ en Anglais) est un segment court dans une séquence, il est continu et non ambigu. Il peut représenter une structure plus complexe lorsque lui-même est composé de différents "motifs" qui peuvent être plus ou moins éloignés les uns des autres et sa définition peut comporter des

exclusions ou des associations de "motifs". Les motifs sont souvent recherchés dans des séquences car ils sont généralement impliqués dans des systèmes de régulation ou ils définissent des fonctions biologiques comme la détermination de la fonction d'une nouvelle séquence (par exemple en localisant un ou plusieurs motifs répertoriés dans des bases de motifs), l'identification dans une séquence nucléique de régions codantes, ou bien l'extraction à partir des banques de données (par exemple extraire des séquences possédant le même signal de régulation ou la même signature protéique pour effectuer des études comparatives ultérieures) . (16)

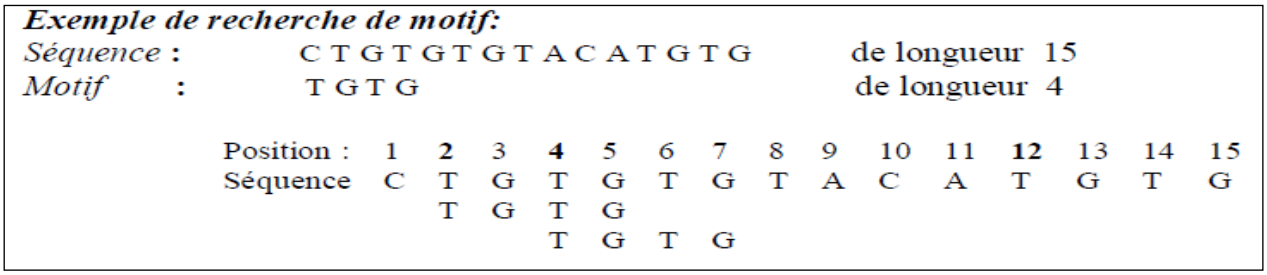

Figure 17 : Exemple de recherche de motif.

2. Découverte de motif

La découverte de motifs joue un rôle clé dans la recherche d'associations, de corrélations et d'autres relations entre les données. De plus, la découverte de motifs peut aider l'indexation des données, la classification.

- **Formalisation du problème**

Dans sa forme la plus simple, le problème de l'extraction de motifs, noté EM, se définit ainsi :

Extraction de motifs communs à un ensemble de séquences.

Données: un ensemble de séquences $\mathcal{S} = \{s_1, \ldots, s_t\}$ $(s_i \in \Sigma^*, 1 \leq i \leq t)$, ainsi qu'une notion de similarité globale.

Problème: trouver toutes les collections de motifs $(m_1 \in s_1, \ldots, m_t \in s_t)$ telles que $m_1, \ldots, m_t$ soient similaires.

La notion de similarité globale (i.e., entre plusieurs motifs) peut être définie de plusieurs manières.

Il peut s'agir d'une notion basée sur un score d'alignement desdits motifs, d'une notion définie en comparant les motifs deux à deux, d'une notion basée sur la similarité entre les motifs et un motif externe ou consensuel.(17)

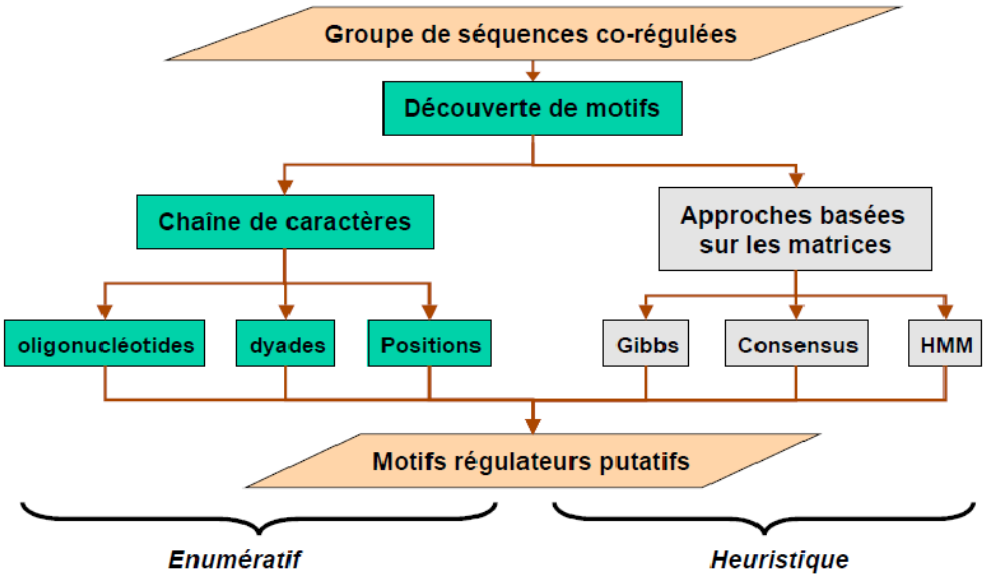

Figure 18 : Différentes approches pour la découverte de motifs.

3. Pourquoi la découverte de motif dans les séquences d'ADN et de protéines?

- Il est clair que l'identification des motifs dans les séquences d'ADN et de protéines fournit des indices importants quant à leur éventuelle réglementation, la fonction ou la structure.

- Identifier particulières motifs d'ADN sera également crucial dans le déchiffrement séquences génomiques.
- A partir d'un génome entièrement séquencé, différentes questions peuvent être abordées par des approches bioinformatiques, en fonction des informations à portée de main:
- Vous voulez savoir si vos gènes préférés possèdent une réglementation particulière élément dans leur promoteur.
- Vous avez identifié un ensemble de gènes co-exprimés et vous voulez savoir si elles sont co-régulées par un facteur de transcription commun.
- Vous avez une collection de modèles et vous voulez trouver des gènes possibles ayant ce motif (c'est à dire des cibles possibles d'un facteur de transcription donné). **(18)**

VII. Représentation de motif biologique :

Pour rechercher un motif dans des séquences personnelles ou dans des banques de données, il va exister différentes façons de l'écrire en fonction de sa complexité. De même, pour découvrir un motif conservé dans un jeu de séquences, le motif identifié sera retourné sous différents formats d'écriture selon le logiciel utilisé.

La définition d'un motif nucléique commence en général par l'analyse d'un alignement multiple de toutes les séquences connues comme étant actives pour la fonction étudiée.

1. Alignement de séquences

La ressemblance de séquences peut également être étudiée au niveau de la séquence entière, et non pas au niveau de quelques motifs. Dans le cas de la recherche ou de l'extraction de motifs, les comparaisons se font localement sur les séquences. Il est question de ressemblance (ou similitude) locale.

Dans le cas de l'étude, non plus de la présence d'un motif biologique

particulier, mais d'une structure globale commune à plusieurs séquences, il est alors question de ressemblance globale. La majorité des méthodes recherchant des similitudes globales entre deux ou plusieurs séquences est basée sur un mécanisme d'alignement des séquences. L'alignement se lit en deux dimensions. Les lignes correspondent aux séquences, tandis que les colonnes correspondent aux positions dans chacune des séquences. Dans une même colonne, deux acides nucléiques ou aminés qui ne se correspondent pas sont alors des substitutions. Selon les choix de représentations, il est possible ou non d'insérer un « trou » dans la colonne, représentant alors une suppression d'un acide dans la séquence (ou une insertion dans les séquences ne contenant pas de trou). La lecture de l'alignement vise à mettre en évidence les parties communes à toutes les séquences. À l'instar des problèmes de la recherche et de l'extraction de motifs, la principale difficulté consiste à modéliser la notion de ressemblance globale, puis à délimiter l'espace des possibles. Ce problème est néanmoins très proche – à plusieurs égards – du problème précédent, et les modélisations de la ressemblance sont souvent les mêmes.

Cela permet de connaître pour chaque base la variabilité à chaque position. L'alignement de ces séquences peut servir à produire une séquence consensus, une table de fréquences ou une matrice de pondération des éléments qui composent le motif. (19)

2. Le consensus et les expressions régulières :

La séquence consensus rend compte de la ou des bases les plus fréquemment rencontrées pour chaque position. La séquence consensus est construite à partir de l'alphabet IUPAC(Figure suivant) ou en retenant une seule base la plus fréquente, pour chacune des positions de la séquence.

These are the official IUPAC-IUB single-letter base codes

Code	Base Description	
G	Guanine	
A	Adenine	
T	Thymine	
C	Cytosine	
R	Purine	(A or G)
Y	Pyrimidine	(C or T or U)
M	Amino	(A or C)
K	Ketone	(G or T)
S	Strong interaction	(C or G)
W	Weak interaction	(A or T)
H	Not-G	(A or C or T)
B	Not-A	(C or G or T)
V	Not-T (not-U)	(A or C or G)
D	Not-C	(A or G or T)
N	Any	(A or C or G or T)

Figure 19 : Alphabets IUPAC.

Dans le cas des séquences très spécifiques, cette simple séquence consensus suffit pour décrire de manière satisfaisante un motif simple. Malheureusement, dans la plupart des cas comme pour les facteurs de transcription, elle ne suffit pas pour identifier les sites biologiquement actifs car elle n'est pas forcément celle qui est le plus souvent rencontrée comme signal. Au pire elle peut elle-même ne pas exister en tant que signal. Ceci est du au fait que celle-ci ne représente qu'un résumé de toutes les séquences.

Pour limiter ce problème, la possibilité d'accepter plusieurs bases ou d'en exclure à certaines positions du motif peut être incorporée dans sa représentation.

Dans ce cas on peut utiliser des expressions régulières pour écrire un motif en utilisant par exemple la grammaire PROSITE résumée dans le tableau ci-dessous:

A	Une seule lettre représente la base biologique, dans notre exemple l'Adénine représentée par un A.
X	X signifie n'importe quelle base biologique.
[ATC]	Une liste représente la possibilité de trouver une des bases qui composent la liste pour une position, pour cet exemple : A, T ou C
{T}	Liste d'exclusion : pas de T à cette position
[AT](2)	pour cet exemple : deux A ou T
x(0,1)	Entre 0 et 1 base quelconque.
<C	pour cet exemple : C au début de la séquence
T>	pour cet exemple : T à la fin de la séquence
-	Élément séparateur du motif

Table 2: Expression régulières.

Exemple : <[AT]-G-x(3)-A-T>, ce motifs signifie qu'en première position il y a un A ou un T, puis un G puis 3 bases inconnues puis A puis T en dernière position.

Ainsi, les expressions régulières permettent de visualiser uniquement les bases possibles à chaque position dans le motif et ne prennent pas en compte les statistiques représentant la variation des bases à chaque position.

3. La matrice de pondération : PWMs

A partir d'un ensemble se séquences biologiques, on peut construire une matrice qui reflète les résidus préférés à chaque position d'un motif de longueur fixe.

- Chaque colonne représente une position
- Chaque ligne représente un résidu ou nucléotide (20 lignes pour les protéines, 4 lignes pour l'ADN)

Les cellules indiquent la fréquence de chaque résidu dans chaque position de l'alignement multiple.

Capturer la fréquence de chaque lettre à chaque position dans le modèle

afin qu'on puisse dire à quel point un site potentiel correspond au modèle.

Il existe trois types de matrices.

- **Positions Count Matrice (PCM)** : enregistre le nombre d'occurrence de chaque résidu ou nucléotides à chaque position
- **Positions Frequency Matrice (PFM)** : enregistre la fréquence dépendant de la position de chaque résidu ou nucléotide
- **Positions Weights Matrice (PWM)** ou Position SpecificScoring Matrices (PSSM): PWM est calculés à partir PFM.
- **PFM corrigé (PFM')** : Les pseudo-poids (pseudo-weight) ont été introduites par Hertz &Stormo (1999) pour tenir compte du petit nombre de séquences utilisées pour construire la matrice PWM.

Site (1)	A	G	A	T	C	C	A	T
Site (2)	T	G	A	C	T	G	A	T
Site (3)	T	C	A	T	C	G	T	T
Site (4)	A	G	A	T	T	G	A	T
Site (5)	T	C	A	A	G	G	A	T
Site (6)	T	G	A	T	C	G	A	C
Site (7)	A	A	A	T	C	G	A	T
consensus:	T	G	A	T	C	G	A	T
IUPAC consensus:	W	V	A	H	B	S	W	Y

	1	2	3	4	5	6	7	8
A	3	1	7	1	0	0	6	0
C	0	2	0	1	4	1	0	1
G	0	4	0	0	1	6	0	0
T	4	0	0	5	2	0	1	6

Table 3 : Table de PFM.

	1	2	3	4	5	6	7	8
A	0.43	0.14	1	0.14	0	0	0.86	0
C	0	0.29	0	0.14	0.71	0.14	0	0.14
G	0	0.57	0	0	0.14	0.86	0	0
T	0.57	0	0	0.71	0.29	0	0.14	0.86

	1	2	3	4	5	6	7	8
A	0.54	-0.58	1.39	-0.58	-4.27	-4.27	1.23	-4.27
C	-4.27	0.15	-4.27	-0.58	1.04	-0.58	-4.27	-0.58
G	-4.27	0.82	-4.27	-4.27	-0.58	1.23	-4.27	-4.27
T	0.82	-4.27	-4.27	1.04	0.15	-4.27	-0.58	1.23

$$W_{i,j} = \ln\left(\frac{f_{i,j}}{p_i}\right)$$

$$f_{i,j} = \frac{n_{i,j} + p_i k}{\sum_{r=1}^{A} n_{r,j} + k}$$

k = pseudo-count
k=0.1

Table 4:Table de PWM .

4. Les Modèles de Markov cachés (HMMs) :

- **Définition :**

Les profiles HMM sont des modèles statistiques (comme les profiles) d'alignements multiples de séquences. Ils tiennent compte de toutes les informations concernant la conservation de chaque position dans chaque colonne d'un alignement en assignant une probabilité représentant les résidus préférés par position, ainsi que les insertions-délétions.

Ce modèle est souvent utilisé pour représenter les modules de régulation : les groupes de motifs qui interagissent. En effet, les modules de cis-régulation impliquent souvent des clusters de sites de fixation pour un ou plusieurs facteurs de transcription.

Un modèle de Markov caché est basé sur un modèle de Markov, sauf qu'on ne peut pas observer directement la séquence d'états : les états sont cachés. Chaque état émet des "observations" qui, elles, sont observables. On ne travaille

donc pas sur la séquence d'états, mais sur la séquence d'observations générées par les états. (20)

- ## Comment un HMM génère-t-il une séquence ?

Le processus de génération d'une séquence de symboles à l'aide d'un HMM consiste à débuter de l'état Begin, puis à se déplacer d'états en états en utilisant les probabilités de transition T. Après chaque transition, la distribution de probabilités de génération G associée à l'état d'arrivée est utilisée pour générer un symbole. Le processus se termine lorsque l'on atteint l'état final End.

On génère ainsi une séquence de symboles $S = s_1 \ldots s_L$, suivant une séquence d'états, ou chemin $C = q_0 \ldots q_L+1$ (où q_0 est l'état Begin et q_L+1 l'état End).

Un HMM définit donc un processus probabiliste non-déterministe, au sens où une même séquence de symboles peut être générée par plusieurs chemins différents.

On comprend alors mieux le nom donné à ce modèle. Le processus de génération est un processus :

- **markovien**, les probabilités de transition et de génération ne dépendent que de l'état actuel et non des états rencontrés précédemment,
- **caché**, car il est impossible de connaître le processus suivi pour la génération d'une séquence de symboles. (21)

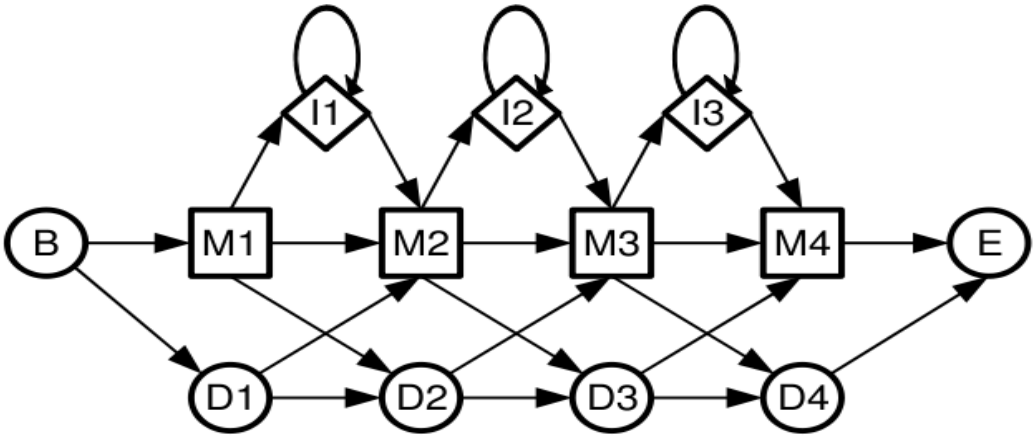

Figure 20 : Structure des HMM profils.

VIII. Les techniques actuelles pour la Découverte de motifs :

Pour traiter le problème de découverte motif, à ce jour différentes approches et outils ont été proposés.la plupart de la littérature antérieure classent ces en deux grands groupes basés sur l'approche combinatoire utilisé dans leur conception:

- fondée sur le mot (basée sur une chaîne) des méthodes qui comptent principalement sur l'énumération exhaustive, c'est à compter et à comparer les fréquences d'oligonucléotides.
- des modèles probabilistes de séquences où les paramètres du modèle sont estimés en utilisant le principe du maximum de vraisemblance ou d'inférence bayésienne.

1. Les méthodes énumératives :

Les algorithmes énumératifs couvrent exhaustivement l'espace de tous les motifs possibles, pour une description spécifique de modèle de motif basées sur le mot garantir l'optimalité globale et ils sont appropriés pour des motifs courts et sont donc utiles pour trouver des motifs dans les génomes eucaryotes où les motifs sont généralement plus courtes que les procaryotes.

Les méthodes basées sur mot peut aussi être très rapide lorsque la mise en œuvre avec les structures de données optimisées telles que les arbres de suffixes et sont un bon choix pour trouver des motifs totalement contrainte, c'est à dire, toutes les instances sont identiques. Toutefois, pour des motifs typiquement des facteurs de transcription qui ont souvent plusieurs positions faiblement restreintes, les méthodes basées sur l'occurrence exacte des mots spécifiques sont trop rigide, elles peuvent être problématiques et le résultat a souvent besoin d'être post-traitées avec un système de clustering. Les méthodes basées sur des mots souffrent également du problème de produire trop de motifs fallacieux.

2. L'approche probabiliste

Implique une représentation du modèle à motifs par une matrice de poids de position (Position Weight Matrices) PWM. Les matrices de poids de position sont souvent visualisées comme un pictogramme dans lequel chaque position est représentée par une pile de lettres, dont la hauteur est proportionnelle au contenu d'informations de cette position.

Les méthodes probabilistes ont l'avantage de nécessiter peu de paramètres de recherche, mais reposent sur des modèles probabilistes des régions régulatrices, qui peuvent être très sensibles à l'égard de petits changements dans les données d'entrée. La plupart des algorithmes développés à partir de l'approche probabiliste sont conçus pour trouver des motifs plus ou plus généraux que sont nécessaires pour les sites de liaison de facteurs de transcription. Par conséquent, ils sont plus appropriés pour la découverte de motif chez les procaryotes, où les motifs sont généralement plus longs que chez les eucaryotes.

Cependant, ces algorithmes ne sont pas garantis pour trouver des solutions globalement optimales, car ils emploient une certaine forme de recherche locale, telles que l'échantillonnage de Gibbs.

La« Expectation Maximisation » (EM) ou algorithmes gloutons qui peuvent converger vers une solution localement optimale**. (22)(23)(24)(25)**

IX. Outils d'analyses des séquences :

- **AlignACE**[5] (Aligns Nucleic Acid Conserved Elements): est un programme qui trouve des éléments de séquences conservées dans un ensemble de séquences d'ADN.
- **MEME**[6] **(Multiple Em for Motif Elicitation) :** est un outil pour découvrir des motifs dans un groupe d'ADN liée ou séquences de protéines.
- **QuickScore7** : Baser sur un algorithme d'une recherche exhaustive qui estime la probabilité de rare ou fréquent mot dans un texte génomique.
  Algorithmes statistiques reposent sur l'hypothèse de base que les nucléotides dans le génome d'un organisme sont générés aléatoirement selon un modèle de probabilité. Deux modèles communs de la génomique sont le modèle de Bernoulli et le modèle de Markov. Selon le modèle, caractéristiques différentes - par exemple l'espérance, la variance, le Z-score et la pvaleur d'un motif donné peut être calculée.

- **SeSiMCMC**[8] **: (The Sequence Similarities by Markov Chain Monte Carlo)**

Algorithme trouve des motifs d'ADN de longueur inconnue et de structure complexe, tels que les répétitions directes ou palindromes avec espaces dans le milieu dans un ensemble de séquences d'ADN.

- **YMF (Yeast Motif Finding) :** Utilise un algorithme de recherche exhaustive pour trouver des motif avec un z-scores élevé.
- **ANN-Spec (Artificial Neural Network Specificity):** Modéliser la spécificité de liaison à l'ADN de un facteur transcription en utilisant une matrice de poids.

[5]http://atlas.med.harvard.edu/
[6]http://meme.sdsc.edu/meme/intro.html
[7]http://algo.inria.fr/dolley/QuickScore/
[8] http://favorov.bioinfolab.net/SeSiMCMC/

- **Consensus[9] :** Ce programme détermine motifs consensus dans des séquences non alignées. L'algorithme est basé sur une représentation matricielle d'un motif consensus.

- **GLAM[10] (Gapless Local Alignment of Multiple sequences):**

GLAM est un programme de détection des motifs fonctionnels partagés par un ensemble de séquences nucléotidiques. GLAM tente pour trouver ces motifs en obtenant la meilleure possible sans gaps, d'alignement multiple de segments de la séquence.

[9] http://stormo.wustl.edu/consensus/
[10] http://zlab.bu.edu/glam/

X. Conclusion

Avec le déluge courant des données biologiques, les méthodes informatiques sont devenues indispensables aux investigations biologiques. A l'origine développée pour l'analyse des séquences biologiques, la bioinformatique couvre maintenant un large éventail de domaines comprenant la biologie moléculaire, génomique et l'étude de séquence nucléique et protéique. Dans ce chapitre, nous avons fourni une introduction sur la bioinformatique et une vue d'ensemble de l'état actuel de ce domaine. En particulier, nous avons montré les différentes pratiques d'analyse une séquence et de bases de données biologiques qui sont généralement employés.

Nous avons en particulier présenté la nature de l'information (séquence) utilisée par un bioinformaticien et les domaines d'utilisation de la bioinformatique. Parmi les thèmes abordés, la découverte de motif dans un ou plusieurs séquences constitue pour certaines pratiques une étape primordiale pour sa progression. Pour ce thème, nous avons consacré un chapitre entier, le suivant.

Chapitre III

L’optimisation par les algorithmes génétiques

I. Introduction

Les problèmes d'optimisation occupent actuellement une place très importante dans le domaine de la recherche. L'évolution des techniques informatiques a permis de dynamiser les recherches dans ce domaine. La résolution d'un problème d'optimisation consiste à explorer un espace de recherche afin de maximiser (ou minimiser) une fonction donnée. Les complexités (en taille ou en structure) relatives de l'espace de recherche et de la fonction à maximiser conduisent à utiliser des méthodes de résolutions radicalement différentes. En première approximation, on peut dire qu'une méthode déterministe est adaptée à un espace de recherche petit et complexe et qu'un espace de recherche grand nécessite plutôt une méthode de recherche stochastique (recuit simulé, algorithme génétique, ...).

L'usage d'un algorithme génétique est adapté à une exploration rapide et globale d'un espace de recherche de taille importante et est capable de fournir plusieurs solutions. Dans le cas où l'ensemble des solutions admissibles est complexe (i.e. il est difficile d'isoler une solution admissible), l'admissibilité peut être rendue intrinsèque à la représentation choisie ou intégrée à la génération des chromosomes (mutation, croisement) ou à la fonction à optimiser (on attribue une mauvaise adaptation à une solution non admissible).

II. Problème d'optimisation :

Un problème d'optimisation se définit comme la recherche du minimum ou du maximum (de l'optimum donc) d'une fonction donnée (26)

Il est défini par un espace d'état, une ou plusieurs fonction(s) objectif(s) et un ensemble de contraintes. Mathématiquement parlant, un problème d'optimisation se présentera sous la forme suivante :

$$\begin{cases} minimiser\ (maximiser)\ f\ (\overrightarrow{x}) & \text{(fonction à optimiser)} \\ avec\ \overrightarrow{g}\ (\overrightarrow{x}) \leq 0 & (m \text{ contraintes d'inégalité}) \\ et\ \overrightarrow{h}\ (\overrightarrow{x}) = 0 & (p \text{ contraintes d'égalité}) \\ \overrightarrow{Min} \leq \overrightarrow{x} \leq \overrightarrow{Max} & (n\ contraintes\ de\ domaines) \end{cases}$$

On a : $\vec{x}\epsilon\mathbb{R}^n, \vec{g}(x)\epsilon\mathbb{R}^m,\ \vec{h}(x)\epsilon\mathbb{R}^p$

Ici, les vecteurs $\vec{g}(x)$ et $\vec{h}(x)$ représentent respectivement m contraintes d'inégalité et p contraintes d'égalité.

III. Vocabulaires et définitions :

1. Fonction à optimiser :

C'est le nom donné à la fonction , c'est la fonction f qu'on cherche à optimiser .ou le but à atteindre pour le décideur. Elle définit un espace de solutions potentielles au problème.

2. Variables de décision :

Elles sont regroupées dans le vecteur $\vec{x}$ qui correspond à l'ensemble des variables du problème. C'est à l'utilisateur de définir les variables du problème. Il peut avoir intérêt à faire varier un grand nombre de paramètres pour augmenter les degrés de liberté de l'algorithme afin de découvrir des solutions nouvelles. Ou bien, s'il a une vue suffisamment précise de ce qu'il veut obtenir, il peut limiter le nombre de variables à l'essentiel.

Les variables peuvent être de natures diverses. Par exemple, pour un composant électronique il peut s'agir de sa forme et de ses dimensions géométriques, des matériaux utilisés, des conditions de polarisation, etc. Nous désignerons par $x_1, x_2 \ldots x_n$ les n variables du problème. Celles-ci peuvent être réelles, complexes, entières, booléennes, etc….. (27)

3. Ensemble des contraintes :

Elle définit des conditions sur l'espace d'état que les variables doivent satisfaire. Ces contraintes sont souvent des contraintes d'inégalité (≤, ≥) ou d'égalité (=) et permettent en général de limiter l'espace de recherche.

4. Minimum globale :

Un « point » $\overrightarrow{x^*}$ est appelé minimum global de la fonction $\boldsymbol{f}$ si :

$$\forall\, \vec{x}, \vec{x} \neq \overrightarrow{x^*} \implies \boldsymbol{f}(\overrightarrow{x^*}) < f(\vec{x})$$

Exemple : M3 dans la figure 21

5. Minimum local fort :

$\overrightarrow{x^*}$est un minimum local fort de la fonction $\boldsymbol{f}$ si :

$$\forall\, \vec{x} \in \boldsymbol{v}(\overrightarrow{x^*}), \vec{x} \neq \overrightarrow{x^*} \implies \boldsymbol{f}(\overrightarrow{x^*}) < f(\vec{x})$$

Où$\boldsymbol{v}(\overrightarrow{x^*})$représente le voisinage de $\overrightarrow{x^*}$

Exemple **:** M2 et M4 dans la figure 21

6. Minimum local faible :

$\overrightarrow{x^*}$est un minimum local faible de la fonction $\boldsymbol{f}$ si :

$$\forall\, \vec{x} \in \boldsymbol{v}(\overrightarrow{x^*}), \vec{x} \neq \overrightarrow{x^*} \implies \boldsymbol{f}(\overrightarrow{x^*}) < f(\vec{x})$$

Exemple **:** M1 dans la figure 21

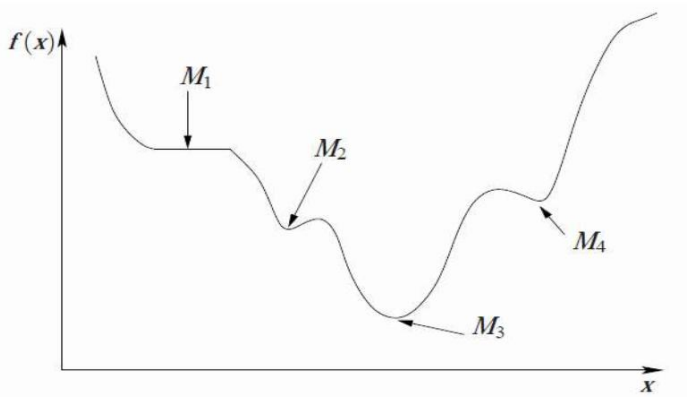

Figure 21 : Exemple des différents miniums.

7. Une méthode d'optimisation :

Elle recherche le point ou un ensemble de points de l'espace des états possibles qui satisfait au mieux un ou plusieurs critère(s). Le résultat est appelé valeur optimale ou optimum.

IV. Classification des problèmes d'optimisation :

Il existe plusieurs critères de classification de types d'optimisation comme montre le schéma suivant (28):

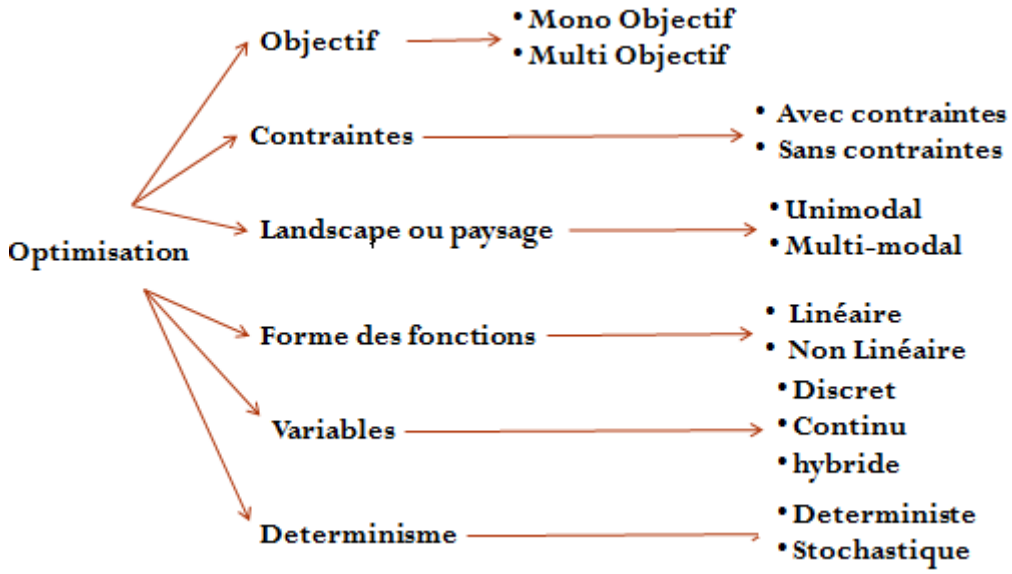

Figure 22 : Problèmes d'optimisation.

On peut classer les différents problèmes d'optimisation en fonction de leurs caractéristiques :

1. Nombre de variables de décision :

- Une ⇒mono variable.
- Plusieurs ⇒multi variable.

2. Type de variable de décision :

- Nombre réel continu ⇒ continu.
- Nombre entier ⇒ entier ou discret.
- Permutation sur un ensemble fini de nombres ⇒combinatoire.

3. Type de fonction objective :

- Fonction linéaire des variables de décision ⇒linéaire.

- Fonction quadratique des variables de décision ⇒quadratique.
- Fonction non linéaire des variables de décision ⇒non linéaire.

4. Formulation de problème :

- Avec contraintes ⇒contraint.
- Sans contraintes ⇒non contraint.

V. Les problèmes d'optimisation mono-objectifs

Lorsqu'un seul objectif (critère) est donné, le problème d'optimisation est mono-objectif. Dans ce cas la solution optimale est clairement définie, c'est celle qui a le coût optimal (minimal, maximal). De manière formelle, à chaque instance d'un tel problème est associé un ensemble A des solutions potentielles respectant certaines contraintes et une fonction d'objectif :

$$\boldsymbol{f : A \rightarrow B}$$

qui associe à chaque solution admissible s ∈ A une valeur de($\boldsymbol{s}$). Résoudre du problème d'optimisation consiste à trouver la solution optimale $\boldsymbol{s^*} \in \boldsymbol{A}$ qui optimise (minimise ou maximise) la valeur de la fonction objective$\boldsymbol{f}$. (29)

VI. Optimisation combinatoire

On qualifie généralement de « combinatoires » les problèmes dont la résolution se heurte à une explosion du nombre de combinaisons à explorer. C'est le cas par exemple lorsque l'on cherche à concevoir un emploi du temps : s'il y a peu de cours à planifier, le nombre de combinaisons à explorer est faible et le problème sera très rapidement résolu ; cependant, l'ajout de quelques cours seulement peut augmenter considérablement le nombre de combinaisons à explorer de sorte que le temps de résolution devient excessivement long.

VII. Les méthodes d'optimisation

Il existe deux classes de méthodes de résolution pour traiter les problèmes d'optimisations les méthodes exactes dédiées à résoudre optimal ment les petites instances et les méthodes approchées : les heuristiques et en particulier les métaheuristiques (génériques) permettant d'approximer les meilleures solutions sur les plus grandes instances. Enfin, le choix de la méthode de résolution à mettre en œuvre dépendra souvent de la complexité du problème. En effet, suivant sa complexité, le problème pourra ou non être résolu de façon optimale . Si le problème est de petite taille, alors un algorithme exact permettant de trouver la solution optimale peut être utilisé (Branch & Bound, programmation dynamique...).

Malheureusement, ces algorithmes par nature énumératifs, souffrent de l'explosion combinatoire et ne peuvent s'appliquer à des problèmes de grandes tailles (même si en pratique la taille n'est pas le seul critère limitant). Dans ce cas, il est nécessaire de faire appel à des heuristiques permettant de trouver de bonnes solutions approchées. Parmi ces heuristiques, on trouve les métaheuristiques qui fournissent des schémas de résolution généraux permettant de les appliquer potentiellement à tous les problèmes.

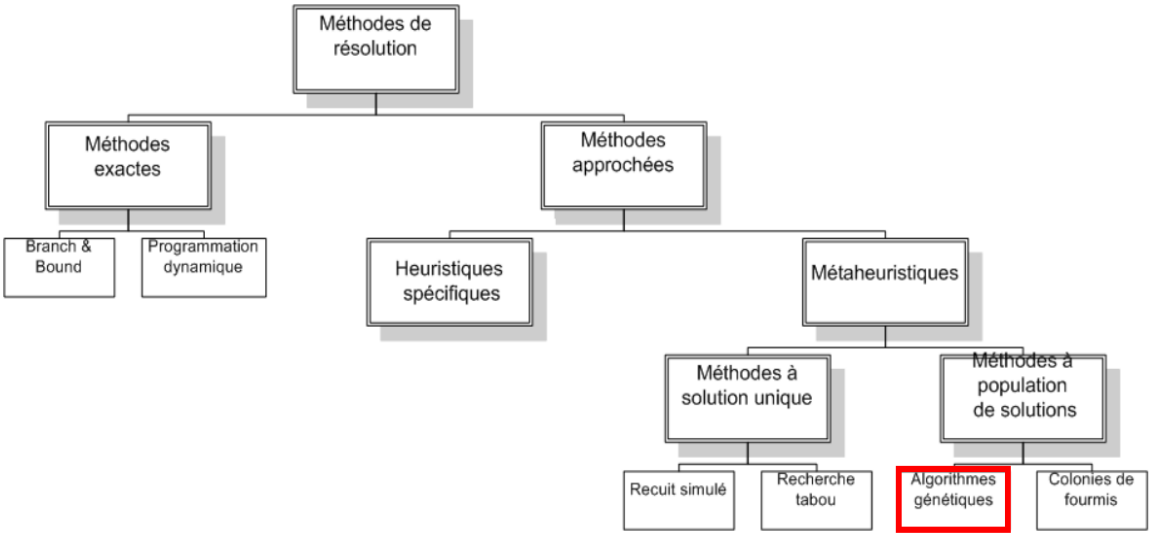

Figure 23 : Les méthodes d'optimisation

1. Les méthodes exactes :

Les méthodes exactes cherchent à trouver de manière certaine la solution optimale en examinant de manière explicite ou implicite la totalité de l'espace de recherche. Elles ont l'avantage de garantir la solution optimale néanmoins le temps de calcul nécessaire pour atteindre cette solution peut devenir très excessif en fonction de la taille du problème (explosion combinatoire) et le nombre d'objectifs à optimiser.

Ce qui limite l'utilisation de ce type de méthode aux problèmes de petites tailles. Ces méthodes génériques sont : Branch & Bound, et D'autres méthodes sont moins générales, comme :La programmation dynamique, simplexe, La programmation linéaire en nombres entiers, D'autres méthodes sont spécifiques à un problème donné comme l'algorithme de Johnson pour l'ordonnancement.

- **Branch & Bound:**

Est une méthode générique de résolution exacte de problèmes d' optimisation, et plus particulièrement d'optimisation combinatoire. C'est une

méthode constructive de recherche arborescente qui utilise l'énumération implicite basée sur la notion de bornes afin d'éviter l'énumération de larges classes de mauvaises solutions. Elle utilise la stratégie diviser pour régner, en se basant sur deux concepts : le branchement (séparation) qui consiste à partitionner ou diviser l'espace des solutions en sous problèmes pour les optimiser chacun individuellement ; et l'évaluation qui consiste à déterminer l'optimum de l'ensemble des solutions réalisables associé au nœud en question ou, au contraire, de prouver mathématiquement que cet ensemble ne contient pas de solution optimale, la méthode la plus générale consiste à borner le coût des solutions contenues dans l'ensemble.

2. Les méthodes approchées :

Méthodes souvent inspirées de mécanismes d'optimisation rencontrés dans la nature. Elles sont utilisées pour les problèmes où on ne connaît pas d'algorithmes de résolution en temps polynomial et pour lesquels on espère trouver une solution approchée de l'optimum global. Elles cherchent à produire une solution de meilleure qualité possible dictée par des heuristiques avec un temps de calcul raisonnable en examinant seulement une partie de l 'espace de recherche.

Dans ce cas l'optimalité de la solution n'est pas garanti ni l'écart avec la valeur optimal. Parmi ces heuristiques, on trouve les métaheuristiques qui fournissent des schémas de résolution généraux permettant de les appliquer potentiellement à tous les problèmes.

Plusieurs classifications des métaheuristiques ont été proposées, la plupart distinguent globalement deux catégories : celles se basant sur une solution unique et celles se basant sur une population de solution. (29)

- **Les heuristiques**

Une heuristique est une technique qui améliore l'efficacité d'un processus de recherche, en sacrifiant éventuellement l'exactitude ou l'optimalité de la

solution. Pour des problèmes d'optimisation (NP complets) où la recherche d'une solution exacte (optimale) est difficile (coût exponentiel), on peut se contenter d'une solution satisfaisante donnée par une heuristique avec un coût plus faible. Méthode de recherche guidée par des "astuces" qui dépendent du problème traité.

- **Les métaheuristiques**

Si certaines heuristique sont spécifique à un problème, d'autres ont pour vocation de pouvoir être adaptées a divers problèmes. On appelle parfois ces dernières des métaheuristiques ou encore des heuristiques générales. Une métaheuristique est des méthodes de recherche indépendantes du problème traité, elle est vue comme un ensemble de concepts qui peuvent être utilisés pour définir des méthodes heuristiques qui peuvent être appliquées à un large éventail de problèmes différents.

En d'autres termes, une métaheuristique peut être considérée comme un cadre algorithmique général qui peut être appliqué à différents problèmes d'optimisation avec des modifications relativement peu pour les rendre adaptés à un problème spécifique. Les métaheuristiques sont en général non-déterministes et ne donnent aucune garantie d'optimalité. Méthodes de recherche indépendantes du problème traité.

- **Métaheuristique à base de solution unique**

Travaillent sur un seul point de l'espace de recherche à un instant donné en commençant avec une solution initiale puis de l'améliorer itérativement en choisissant une nouvelle solution dans son voisinage

- **Le recuit simulé :**

La méthode du recuit simulé est basée sur une analogie avec le processus physique de recuit des matériaux cristallins. Ce dernier consiste à amener un élevé à forte température.

- **La recherche Tabou :**

La recherche Tabou est une méthode de recherche locale. A chaque itération, l'algorithme examine le voisinage V d'un point x et retient le meilleur voisin x' tel que : $\forall x'' \in V(x)$, f(x') est meilleur que f(x") et x' $\notin$ à la liste tabou, sauf si f(x') est meilleur que f(x) : c'est le phénomène d'aspiration. On note que x' est retenu même si f(x') n'est pas meilleur que f(x)

- **Métaheuristiques à base de population de solutions**

Travaillent sur un ensemble de points de l'espace de recherche en commençant avec une population de solution initiale puis de l'améliorer au fur et à mesure des itérations. L'intérêt de ces méthodes est d'explorer un très vaste espace de recherche et d'utiliser la population comme facteur «de diversité» de plus elle sont très adaptées et très largement utilisées .(29) **(**(30)

- **Colonie de fourmis**

Les algorithmes à base de colonies de fourmis ont été introduits par **Dorigo**. Une des applications principales de la méthode originale était le problème du voyageur de commerce et depuis elle a considérablement évolué. Cette nouvelle métaheuristique imite le comportement de fourmis cherchant de la nourriture. A chaque fois qu'une fourmi se déplace, elle laisse sur la trace de son passage une odeur (la phéromone).

- **Les algorithmes évolutionnaires**

Les algorithmes évolutionnaires sont inspirés des concepts issus du Lamarckisme, Darwinisme et du mutationnisme. Les algorithmes évolutionnaires se caractérisent par :

- Une représentation spécifique des solutions potentielles du problème.
- Un ensemble d'individus formant une population permettant de mémoriser les résultats à chaque étape du processus de recherche.
- Un processus de création aléatoire d'un individu. Cette caractéristique offre une capacité exploratoire importante à la méthode.

- Un ensemble d'opérateurs de modification permettant de créer de nouvelles solutions à partir des informations mémorisées. Ces opérateurs offrent une capacité de recherche locale à la méthode.
- Une heuristique de notation qui représente la sélection effectuée par l'environnement.
- Une heuristique de sélection.
- Un critère d'arrêt de l'algorithme.

On distingue plusieurs types d'algorithmes évolutionnaires :

- **Les algorithmes génétiques** : sont inspirés des mécanismes de l'évolution naturelle.
- **La programmation génétique** : est une extension des algorithmes génétiques dans laquelle les individus sont des programmes. Le génotype d'un individu est constitué d'un alphabet et se présente sous forme arborescente.
- **Les systèmes de classifieur** : sont des mécanismes d'apprentissage basés sur un ensemble de règles condition/action. Chaque règle est notée en fonction du résultat de l'action produite et un algorithme génétique est utilisé pour générer de nouvelles règles
- **Les stratégies d'évolution** : sont des algorithmes itératifs dans lesquels un parent génère un enfant. Le meilleur des deux survit et devient le parent de la génération suivante. (31)

✓ Avantages

- Ces méthodes sont applicables dans la plupart des problèmes d'optimisation : multimodaux, non continus, contraints, bruités, multiobjectifs, dynamiques, etc….
- Elles n'exigent pas d'hypothèse par rapport à l'espace d'état.
- Elles sont flexibles par rapport aux nouvelles contraintes et nouveaux critères à prendre en compte.
- Les résultats sont en général exploitables et interprétables par le décideur.

✓ Inconvénients

- Elles n'offrent aucune garantie de trouver l'optimum en un temps fini. Mais cela est vrai pour toutes les méthodes d'optimisation globales.
- Leur base théorique est insuffisante.

Le réglage des paramètres est largement inspiré de l'essai/erreur sauf pour les stratégies d'évolution qui sont auto-adaptatives.

VIII. Les algorithmes génétiques

Les algorithmes génétiques (AG) sont des méthodes utilisées dans les problèmes d'optimisation. Les AG tirent leur nom de l'évolution biologique des êtres vivants dans le monde réel. Ces algorithmes cherchent à simuler le processus de la sélection naturelle dans un environnement défavorable en s'inspirant de la théorie de l'évolution proposée par C. Darwin. Dans un environnement, « les individus » les mieux adaptés tendent à vivre assez longtemps pour se reproduire alors que les plus faibles ont tendance à disparaître.

Par analogie avec l'évolution naturelle, les AG font évoluer un ensemble de solutions candidates, appelé une « population d'individus ». Un « individu » n'est autre qu'une solution possible du problème à résoudre. Chaque individu de cette population se voit attribuer une fonction appelée fonction d'adaptation (fitness) qui permet de mesurer sa qualité ou son poids ; cette fonction d'adaptation peut représenter la fonction objectif à optimiser. Ensuite, les meilleurs individus de cette population sont sélectionnés, subissent des croisements et des mutations et une nouvelle population de solutions est produite pour la génération suivante.

Ce processus se poursuit, génération après génération, jusqu'à ce que le critère d'arrêt soit atteint, comme par exemple le nombre maximal de générations

1. Principe de base d'un AG standard

Un AG standard nécessite en premier le codage de l'ensemble des paramètres du problème d'optimisation en une chaîne de longueur finie. Le principe d'un AG est simple, il s'agit de simuler l'évolution d'une population d'individus P jusqu'à un critère d'arrêt. On commence par générer une population initiale d'individus P (solutions) de façon aléatoire. Puis, à chaque génération , des individus sont sélectionnés, cette sélection est effectuée à partir d'une fonction objectif appelée fonction d'adaptation (fitness) . Puis, les opérateurs de croisement et de mutation sont appliqués et une nouvelle population est créée. Ce processus est itéré jusqu'à un critère d'arrêt. Le critère le plus couramment utilisé est le nombre maximal de générations que l'on désire effectuer. La figure (24) présente le principe de l'AG standard.

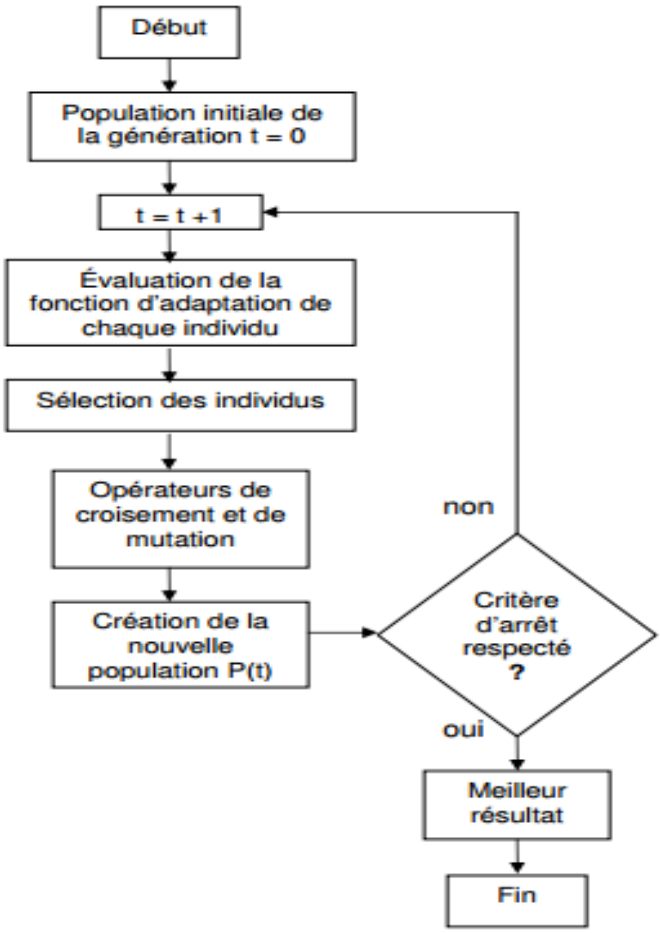

Figure 24 : Organigramme d'un AG standard.

- **Codage :**

Le codage des éléments de la population. C'est à ce moment que l'on détermine les structures de données utiles. Bien entendu cette étape suit la modélisation du problème posé. La réussite des algorithmes génétiques est issue de la qualité de leur codage. Au début, Le codage binaire est le code le plus utilisé (Goldberg, 1989), l'inconvénient majeur du code binaire étant que deux points proches dans l'espace des variables (voir la colonne 1 du Tableau) ne sont pas nécessairement codés par deux chaînes de bits voisines (colonne 2 du Tableau) On remédie en général à ce problème en utilisant le codage de Gray qui. (33)

- **Population initial :**

Le premier pas dans l'implantation des algorithmes génétiques est de créer une population d'individus initiaux P (solutions) de façon aléatoire. En effet, les algorithmes génétiques agissent sur une population d'individus, et non pas sur un individu isolé. Par analogie avec la biologie, chaque individu de la population est codé par un chromosome ou génotype (Holland, 1975). Une population est donc un ensemble de chromosomes. Chaque chromosome code un point de l'espace de recherche. L'efficacité de l'algorithme génétique va donc dépendre du choix du codage d'un chromosome. (30)

- **Evolution de population :**

Pour calculer le coût d'un point de l'espace de recherche, on utilise une fonction d'évaluation. L'évaluation d'un individu ne dépendant pas de celle des autres individus, le résultat fournit par la fonction d'évaluation va permettre de sélectionner ou de refuser un individu pour ne garder que les individus ayant le meilleur coût en fonction de la population courante : c'est le rôle de la fonction fitness. Cette méthode permet de s'assurer que les individus performants seront conservés, alors que les individus peu adaptés seront progressivement éliminés de la population.

- **Sélection**

La sélection a pour objectif d'identifier les individus qui doivent se reproduire. Cet opérateur ne crée pas de nouveaux individus mais identifie les individus sur la base de leur fonction d'adaptation, les individus les mieux adaptés sont sélectionnés alors que les moins bien adaptés sont écartés . La sélection doit favoriser les meilleurs éléments selon le critère à optimiser (minimiser ou maximiser). Ceci permet de donner aux individus dont la valeur est plus grande une probabilité plus élevée de contribuer à la génération suivante.

Il existe plusieurs méthodes de sélection : (**32**)

- La méthode de la "loterie biaisée" (roulette Wheel) de **GoldBerg**,
- La méthode "élitiste",
- La sélection par tournois,
- La sélection universelle stochastique.

Les plus connues étant la « roue de la fortune » et la « sélection par tournoi » :

- **La roue de la fortune :**

Est la plus ancienne où chaque individu, la probabilité d'être sélectionné est proportionnelle à son adaptation au problème ($\boldsymbol{Fitness}\ (\boldsymbol{j})$). Afin de sélectionner un individu, on utilise le principe de la roue de la fortune biaisée. Cette roue est une roue de la fortune classique sur laquelle chaque individu est représenté par une portion proportionnelle à son adaptation. On effectue ensuite un tirage au sort homogène sur cette roue.

La probabilité de sélection d'un individu ($\boldsymbol{j}$) s'écrit :

$$\boldsymbol{Prob(j)} = \frac{\boldsymbol{Fitness(j)}}{\sum_{\boldsymbol{j=1}}^{\boldsymbol{J_{max}}} \boldsymbol{Fitness(j)}}$$

Les individus possédant une plus grande fonction d'adaptation ayant plus de chance d'être sélectionnés.

- **Sélection par tournoi :**

A chaque fois qu'il faut sélectionner un individu, la « sélection par tournoi » consiste à tirer aléatoirement (k) individus de la population, sans tenir compte de la valeur de leur fonction d'adaptation, et de choisir le meilleur individu parmi les k individus. Le nombre d'individus sélectionnés a une influence sur la pression de sélection, lorsque k = 2, la sélection est dite par « tournoi binaire ».

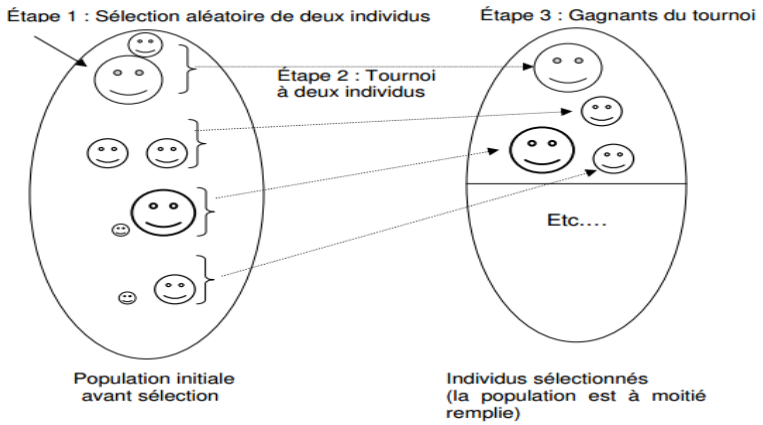

Figure 25: Représentation d'une sélection par tournoi d'individus pour un critère de maximisation.

- **Croisement**

Le croisement permet de créer de nouvelles chaînes en échangeant de l'information entre deux chaînes . Le croisement s'effectue en deux étapes. D'abord les nouveaux éléments produits par la reproduction sont appariés, ensuite chaque paire de chaînes subit un croisement comme suit : un entier k représentant une position sur la chaîne est choisi aléatoirement entre 1 et la longueur de chaîne (l) moins un $(l-1)$. Deux nouvelles chaînes sont créées en échangeant tous les caractères compris entre les positions k +1 et $l$inclusivement.

L'exemple suivant montre deux chaînes de $\boldsymbol{longueur = 14}$ appartenant à la population initiale. Les deux nouvelles chaînes :

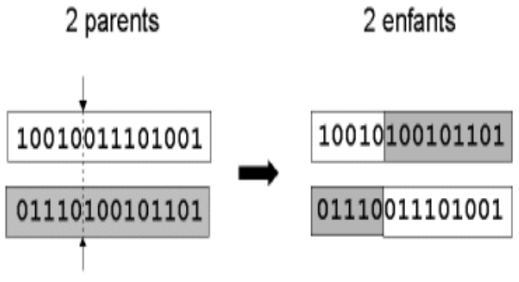

Figure 26:Croisement avec 1 point

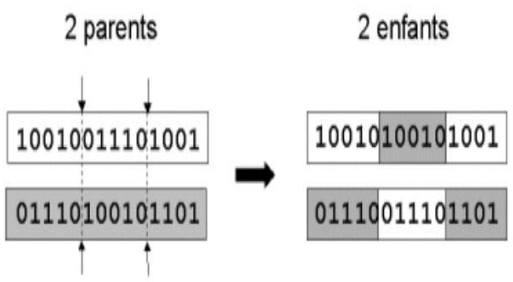

Figure 27 : Croisement avec 2 points

- **Mutation**

La mutation est exécutée seulement sur une seule chaîne. Elle représente la modification aléatoire et occasionnelle de faible probabilité de la valeur d'un caractère de la chaîne, pour un codage binaire cela revient à changer un 1 en 0 et vice versa (figure 28). Cet opérateur introduit de la diversité dans le processus de recherche des solutions et peut aider l'AG à ne pas stagner dans un optimum local.

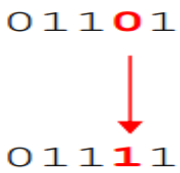

Figure 28 : Représentation d'une mutation de bits dans une chaîne.

- **Elitisme**

A la création d'une nouvelle population, il y a de grandes chances que les meilleurs chromosomes soient perdus après les opérations d'hybridation et de mutation. Pour éviter cela, on utilise la méthode d'élitisme. Elle consiste à copier un ou plusieurs des meilleurs chromosomes dans la nouvelle génération. Ensuite, on génère le reste de la population selon l'algorithme de reproduction usuel. Cette méthode améliore considérablement les algorithmes génétiques, car elle permet de ne pas perdre les meilleurs solutions.

IX. Conclusion

Dans ce chapitre, nous avons essayé de rassembler une introduction sur les problèmes d'optimisation, tout en montrant la manière de définir un problème d'optimisation, ses contraintes, ses objectifs ainsi que les méthodes utilisées pour le résoudre.

Nous avons énuméré les méthodes utilisées dans le domaine de l'optimisation, parmi les métaheuristiques existantes, nous trouvant les algorithmes évolutionnaires qui exercent le principe de la génétique pour résoudre ces problèmes. Puisque notre travail se base sur l'un des algorithmes génétiques nous parleront dans ce chapitre sur ces algorithmes, ses étapes et ses paramètres. Après nous avons présenté dans le chapitre suivant une explication détaillée de l'algorithme génétique utilisé pour la Découverte des motifs et décrit en détail les différentes techniques utilisées pour l'implémentation de notre projet .

Chapitre IV

Les algorithmes génétiques et la découverte de motif

I. Introduction :

Les algorithmes génétique ont été largement utilisés dans le domaine de la bioinformatique, principalement dans l'alignement de séquences (MSA), et se sont révélés assez bien réussi.

Cette technique, peu similaire dans le concept à l'apprentissage de la machine, a beaucoup plus de succès que ce dernier, C'est principalement parce que les algorithmes d'apprentissage automatique effectuent la recherche locale, générant ainsi des solutions qui peuvent être localement optimale, mais sont rarement des solutions globales.

D'autre part, les algorithmes génétiques effectuent une recherche globale sans effectuer une recherche exhaustive.

Bien que les algorithmes génétiques ne garantissent pas toujours une solution optimale, ils ont une meilleure chance de trouver une solution optimale

Les algorithmes génétiques permettent également d'utiliser les fonctions d'adaptation (**fitness**) pour évaluer (le score) les solutions. Ces fonctions d'adaptation (**fitness**) ne doivent pas être constante pour tous les problèmes ; ils peuvent utiliser n'importe quel informations concerne pour évaluer (score) les solutions comprendre l'information biologique, fonctionnelle… etc. Ainsi, ils offrent une flexibilité dans l'évaluation des solutions et offrent une flexibilité de décider comment représenter le problème. Il y a quelques études où les motifs ont été représentés dans diverses formats appropriés pour l'algorithme génétique à l'étude ceux-ci comprennent représentant des motifs que les expressions régulières, les matrices de fréquences de position … etc.

II. Objectif :

L'objectif de ce projet est d'identifier les motifs qui représentent des éléments réglementaires dans les séquences biologiques. L'entrée de l'algorithme se compose de deux ensembles des séquences. Le premier est un ensemble de séquences de **promoteur** qui a la probabilité de contenir les éléments réguliers. Le second est un

ensemble de séquences **d'arrière-plan** représentant une partie aléatoire de l'ADN qui ne peuvent comporter des séquences de **promoteur**.

L'approche suivie dans ce projet pour appliquer un AG dans notre problème (découvert de motif) peut être résumée comme suit:

1. Déterminer une représentation pour les motifs.
2. Évaluer les motifs à l'aide d'une fonction d'adaptation (**fitness**) bien définie.
3. Cluster (***regroupement)*** de la population sur la base de certaines métriques.
4. Exécutez l'algorithme génétique de répéter les étapes ci-dessus jusqu'à ce qu'un motif près de la séquence consensus est identifié.

2. La représentation des Motifs

En général, les algorithmes génétiques offrent la flexibilité de pouvoir utiliser une représentation qui convient le mieux à l'instance du problème et les solutions proposées. En cas de la découverte de motif, diverses représentations ont déjà été utilisées. Il s'agit notamment des expressions régulières, les matrices de fréquences de position (**PFM**), les matrices de poids de position ….

La représentation de motifs utilisés dans ce projet est une combinaison de deux matrices de fréquences de position (PFM) et des matrices de poids de position (PWM).

Cette exemple montre une population (3ensembles des séquences) et la représentation PWM .

Ensemble 1 :	Ensemble 2 :	Ensemble 3 :
TTATGAC	CATTCCCTC	TCTATCAC
GTTATTC	ATCACAATT	GCCTGGTC
TACTTTG	GAACTAAAG	TTCGAAGT
ATTGTGC	GGCGCGAGA	TAGCACAT
GAGACAA	CGTATTCCC	CGAGCGGG
TGCTACC	CGGTTGCTG	CAATATGT
TTACCGG	CTTGGGACC	ACATATTT
TCGGAAC	ATAAAACCT	ACCTCTAC
TCGATCG	CATTCACCG	AATGGATG
GTTGAAC	CGGAACCCG	CGCAAAAA

Les matrices de pondération (**PWM**) des ensemles 1 , 2 et 3

```
>> pwm

ans =

   -0.9015   -0.2207   -0.2207    0.1807    0.1807    0.4663   -0.9015
   -4.6151   -0.2207   -0.2207   -0.9015   -0.2207   -0.2207    0.8697
    0.1807   -0.9015    0.1807    0.1807   -0.9015   -0.2207    0.1807
    0.8697    0.6882    0.1807    0.1807    0.4663   -0.2207   -4.6151

>> pwm

ans =

    0.1807    0.1807    0.1807   -0.2207    0.6882    0.1807    0.4663   -0.9015
    0.1807    0.4663    0.4663   -0.9015   -0.2207   -0.2207   -4.6151    0.1807
   -0.9015   -0.2207   -0.9015    0.1807   -0.2207   -0.2207    0.1807   -0.2207
    0.1807   -0.9015   -0.2207    0.4663   -0.9015    0.1807    0.1807    0.4663

>> pwm

ans =

   -0.2207    0.1807   -0.2207    0.4663   -0.2207    0.4663    0.4663   -0.9015   -0.9015
    0.8697   -4.6151   -0.2207   -0.9015    0.4663   -0.2207    0.8697    0.6882    0.1807
   -0.2207    0.4663   -0.2207   -0.2207   -0.9015    0.1807   -4.6151   -0.9015    0.4663
   -4.6151    0.1807    0.4663    0.1807    0.1807   -0.9015   -4.6151    0.1807   -0.2207

>>
```

3. Evaluation des motifs (fitness)

La fonction de **fitness** utilisée pour évaluer les motifs peut être décrite comme suit:

1. Le motif à l'étude est converti en sa représentation **PWM**
2. Chaque séquence de l'ensemble d'entrée (ensemble de séquences de **promoteur**) et les séquences **d'arrière-plan** est alors considéré. Le score de PWM pour chaque séquence est calculé comme suit:

   a) Le score du **PWM** à chaque offset de la séquence d'entrée est calculée à l'aide de la **PWM** généré ci-dessus

   b) La valeur maximale de score dans l'ordreest considérée comme le meilleur score pour la totalité de la séquence.

   c) Cette valeur est divisée parla valeur maximale possible pour une **PWM** de la taille spécifiée afin de normaliser à une plage de**[0, 1].**

   d) Ce processus est répété pour chaque séquence dans les deux ensembles de données.
3. La meilleure valeur moyenne de match est ensuite calculée pour chacun des deux ensembles de données.

La fitness du motif = L'écart entre les deux valeurs moyennes, calculées comme suit :

Meilleur score moyen pour la séquence promoteur-Meilleur score moyen pour la séquence d'arrière-plan

En calcule les scores de fitness pour l'exemple précédent .

Pour d'ensembles des séquences promoteur et arrière-plan ci-dessous

Les sequences de promoteur

```
TTATGAC
GTTATTC
TACTTTG
ATTGTGC
GAGACAA
TGCTACC
TTACCGG
TCGGAAC
TCGATCG
GTTGAAC
```

Les sequences arrière-plan

```
TCTATCAC
GCCTGGTC
TTCGAAGT
TAGCACAT
CGAGCGGG
CAATATGT
ACATATTT
ACCTCTAC
AATGGATG
CGCAAAAA
```

les valeur d'évaluation pour la population précédente dans la matrice ci-dessous

```
fitScore =

   -0.6860   -0.6776   -0.7681
```

Le fitness score pour l'individu 1 de la population est -0.6860

Le fitness score pour l'individu 2 de la population est -0.6776

Le fitness score pour l'individu 3 de la population est -0.7681

4. Le regroupement de la population (Clustering)

En raison que les éléments de régulation fournissent des sites de liaison pour des facteurs de transcription, il est important de garder à l'esprit qu'il y a toujours une possibilité de la présence de plusieurs motifs dans les régions régulatrices de séquences biologiques. Cela peut poser des problèmes lors de considérant l'approche de l'algorithme génétique standard.

Un algorithme génétique standard fonctionne par l'évolution de la population, tels que la plus forte solution représente toujours la meilleure solution possible. Ainsi les algorithmes génétiques fournir presque toujours une solution unique à la fin du processus, ou très similaire solutions, en cas de multiples solutions. En conséquence, la diversité de la population n'est pas maintenue, ce qui conduit à la perte de l'information sur les divers motifs multiples. Un autre inconvénient est que c'est souvent la seule solution obtenue s'avère être un faux positive. Il est possible pour des solutions avec haut fitness score et biologiquement a aucun sens.

Certaines techniques ont été proposées pour supprimer ces inconvénients. Il s'agit notamment de contrôler le choix des parents sélectionnés pour le croisement, en utilisant des populations réparties dans l'espace, regroupement de la population etc…

La technique de regroupement de la population (clustering) a été choisie pour ce projet. Il s'agit d'utiliser un algorithme de clustering pour diviser la population.

Ces groupements ensuite soumis au processus de **l'accouplement** de

l'algorithme génétique, Un tel intra-groupe **accouplement** permet la diversité de la solution au cours de l'évolution. Après cela, la nouvelle population est encore l'objet de regroupement dans la nouvelle itération. Ceci offre l'avantage de permettre à la solution de passer d'un groupe à l'autre (en fonction de la fitness score), favorisant ainsi un certain inter-cluster accouplement, fournir un autre moyen de maintenir la diversité de la solution.

- ## Leader Algorithm :

Algorithme de leader, un type d'algorithme rapide, il est l'un des plus ancien et classique techniques de regroupement présents. L'algorithme de séparation-rapides offre une beaucoup plus rapide méthode pour regrouper l'espace de solution, par rapport à d'autres méthodes de classification. L'algorithme **Leader** est un des plus simples et des plus rapides méthodes actuelles de regroupement, et le plus approprié quand un grand nombre d'objets sont à être regroupés.

L'algorithme de leader est un algorithme en une passe utilisé pour diviser l'espace en M partitions en calculant la distance **euclidienne** entre toutes les paires d'objets et en la comparant à un seuil **T.**

Il construit des partitions égal au nombre de groupes. Pour chaque cluster un leader est sélectionné (leader), de telle sorte que tous les éléments du cluster sont dans une distance **T** du leader. C'est une approche gourmande où un seul passage est effectué à travers l'espace de solution, et chaque solution est affectée au premier groupe dont le leader est assez proche. Si aucun leader n'est dans la distance T, un nouveau cluster (group) est créé, et la solution est le leader du nouveau cluster.

Pour les besoins de ce projet, une version modifiée de l'algorithme de leader, a été utilisé. La version modifiée est une passe multi-mode .Cette méthode ne nécessite pas de calcul des seuils. Elle commence par un objet initial central, et sur le premier passage, trouve le plus loin de l'objet à partir du central. Cet objet devient le leader du groupe suivant. Le passage suivant consiste à trouver un objet qui est le plus éloigné de son leader correspondant. Cet objet est ensuite devient le leader du prochain cluster.

Ce processus est répété jusqu'à ce que le nombre requis de groupes été formés

Cette méthode offre un avantage sur l'algorithme de leader classique, que aucun seuil est initialisé, mais il est nécessaire d'estimer le nombre des groupes.

Les étapes de cet algorithme sont présentées ci-dessous :

Les variables utilisées sont:

K : le nombre de cluster (groupes),

I : les éléments considérés

L : les leaders de groupe

D : pour la distance entre deux éléments

1. **Démarrez avec l'élément central dans I , place l'objet dans le groupe K=1, et en le fait le leader du groupe K=1**
2. **Pour chaque élément de I, répétez les étapes 2.a et 2.b**
    a) **Pour chaque cluster K existe déjà , trouver la distance entre l'élément et le leader de groupe D(I,LK)**
    b) **Attribuer l'élément I dans le groupe k où D(I,LK) est minimum**
3. **Pour chaque cluster K qui existe déjà, trouver l'élément qui est le plus éloigné de son leader d max=max(D (I,LK)).**
4. **Sélectionnez l'élément ayant une valeur maximale d max, et en le mettant le leader du nouvel groupe, K=K+1,**
5. **Répétez les étapes 2 à 5 jusqu'à ce que le nombre requis des groupes sont créés.**

pour calculer la distance entre deux éléments (PWM) en utilise « **feature vector »**

- **Algorithme à suivre pour calculer le « feature vector »**

Steps for calculating the feature vector

1. Let $index = 0$
2. for all tetra-nucleotides in the following set: {AAAA, AAAC, AAAG, …. , TTTG, TTTT}
   a. $prob_{sum} = 0$, $cnt = 0$
   b. **for** i from 0 to (number of columns in the PFM – 4)
      i. $prob = 1$
      ii. $cnt = cnt + 1$
      iii. **for** each residue of the tetra-nucleotide under consideration
         1. $prob = prob$ * residue value of PFM column (i+residue position) ## Calculate probability for each residue ##
      iv. **end for**
      v. $prob_{sum} = prob_{sum} + p$ ## Calculate total probability for all tetra-nucleotides ##
   c. **end for**
   d. $featureVector[index] = prob_{sum}/cnt$ ## normalize ##
   e. $index = index + 1$
3. **end for**

- **Algorithme à suivre pour calculer la distance entre deux vecteurs**

Le calcul de la distance euclidienne entre les deux vecteurs

1. $sum = 0$, $i = 0$
2. for each element i of the two feature vectors *fv1* and *fv2*
   a. $diff$ = fv1[i] – fv2[i]
   b. $sum = sum + (diff * diff)$
3. $euclideanDistance$ = sqrt(sum)

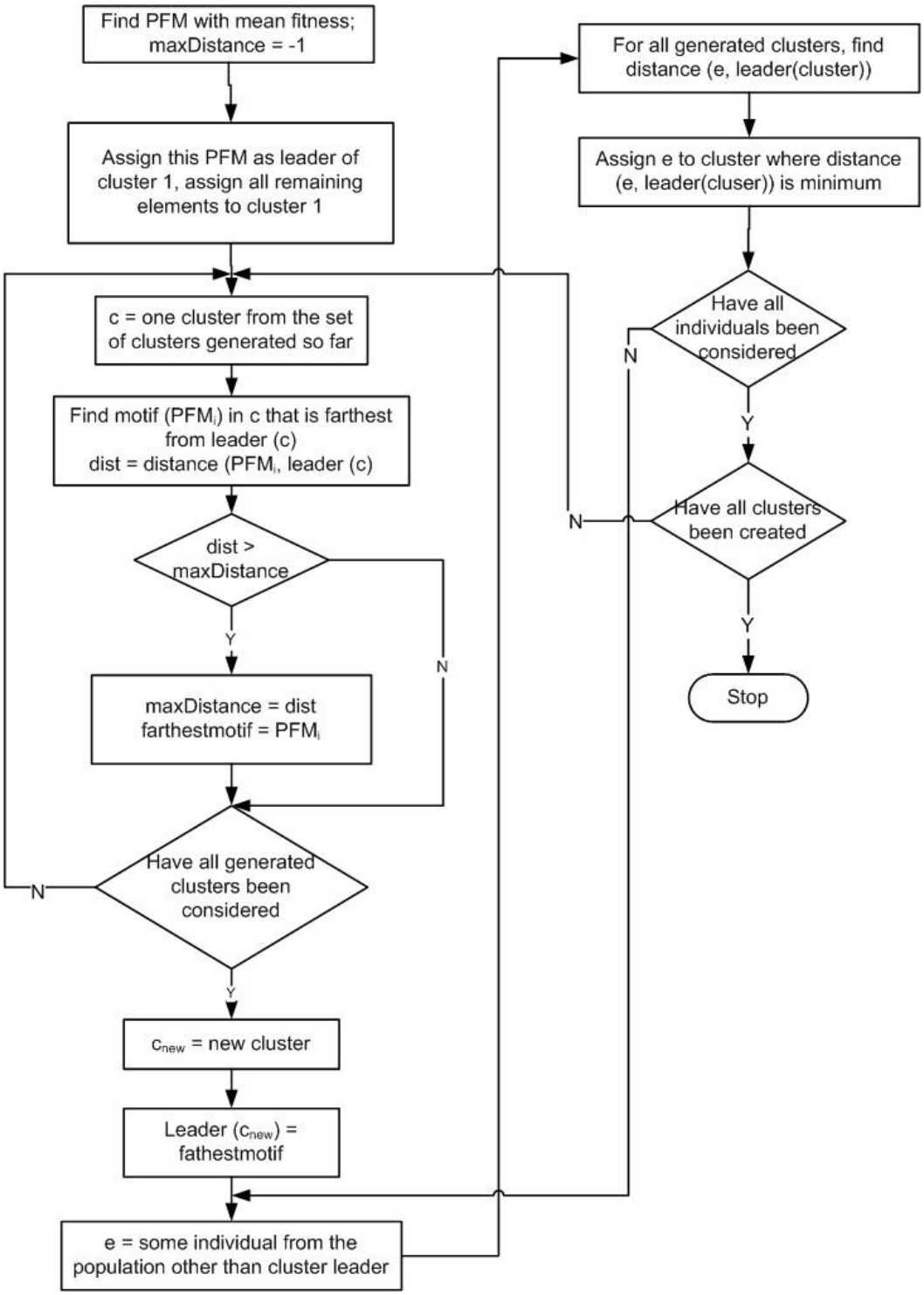

Figure 29 : Les étapes du groupement

Pour le regroupement de notre exemple cette vecteur représente le numéro de groupe de chaque individu (en prend le nombre des clusters = 2)

```
clusterPop =

     1     2     2
```

Le groupe de l'individu 1 est 1
Le groupe de l'individu 2 est 2
Le groupe de l'individu 3 est 2

III. Les étapes de l'AG dans le projet :

Cette section donne une explication de l'algorithme génétique utilisé pour la découverte de motif. Il présente la technique de sélection pour la population, les méthodes utilisées pour produire la nouvelle population, et un flux des étapes de l'algorithme génétique.

L'algorithme génétique utilisé pour la découverte de motif commence par la phase **d'initialisation** cette phase est suivie par un processus de **regroupement de la population**, **l'accouplement** des parents sélectionnes (en utilisant l'élitisme, la mutation et de croisement), et l'évaluation de la nouvelle génération produite.

5. L'initialisation :

La phase d'initialisation consiste à générer de façon **aléatoire** la population initiale.

La population est évaluée à l'aide de la fonction d'adaptation (fitness) décrit précédemment

Cette population est ensuite regroupée en utilisant l'approche décrite précédemment. Après le regroupement la population est autorisée à évoluer dans le

processus de **l'accouplement**. L'accouplement est seulement autorisé dans les groupes qui permettent de maintenir la diversité de la solution.

Le nombre de descendants générés pour chaque groupe est proportionnel à la moyenne score de fitness de toutes les solutions du groupe. Après la phase de reproduction (**l'accouplement**) l'ensemble du processus est à nouveau répété.

6. Sélection :

Pour effectuer les mutations et croisement, les parents sont sélectionnés à l'aide d'une méthode appelée **La roue de la fortune**

La méthode de **La roue** est la plus couramment utilisée pour le processus de sélection. Toutefois, elle ne fonctionne pas très bien quand il existe de grandes différences entre la fitness des objets dans la population. Ainsi, s'il y a des objets qu'ont une fitness élevée comme 80%, et d'autres ont une fitness bien moindre, ces objets de fitness inférieurs auront une très petite chance d'être sélectionnés.

Pour résoudre ce problème, la méthode de sélection utilisée est le **Classement**. Cette méthode un peu modifiée a la méthode de la **roue**. Elle classe tous les objets en fonction de la fitness (de 1 à **N**, où **N** est la taille de la population). Les portions de la roue sont ensuite attribuées en fonction de ces classes, au lieu de la fitness. Cela donne tous ces objets une chance d'être sélectionné en fonction de leurs rangs.

Les étapes de la méthode de sélection selon le Classement sont:

1. Classement de tous les éléments du groupe sur la base de leur fitness, de telle sorte que l'élément avec la fitness la plus basse est de classe1, et celui avec la plus haute fitness devra être une classe N.

   Exemple: Pour 4élémentsavecfitness **0,3, 0,6, 0,9 et 0,1**, leurs classement respectif est: 2, 3, 4et1.
2. Trouver la somme de tous les classes
3. Le pourcentage de la roue attribuée à chaque élément est égale à: **(class *100) /somme des classes**

4. Tourner la roue de la fortune.
5. Déterminer l'élément correspondant à l'emplacement sélectionné.
6. Sélectionnez cet élément

7. Reproduction

Il y a trois méthodes sont utilisées pour l'accouplement :

- **Elitisme :**

Élitisme implique que l'instance avec la meilleure valeur de fitness est autorisée à se propager à la génération suivante, sans aucune modification. Ceci est fait parce que de nombreuses fois la meilleure solution disparaissent en raison de la mutation et le croisement. L'élitisme permet la meilleure solution de la population à avancer tel quelle. Si la solution n'est pas assez bonne, elle se supprime au cours de l'évolution.

Pour l'exemple précédent l'élitisme pour le cluster numéro 2 se fait sur l'individu numéro 2 de la population (ci-dessous)

0.1807	0.1807	0.1807	-0.2207	0.6882	0.1807	0.4663	-0.9015
0.1807	0.4663	0.4663	-0.9015	-0.2207	-0.2207	-4.6151	0.1807
-0.9015	-0.2207	-0.9015	0.1807	-0.2207	-0.2207	0.1807	-0.2207
0.1807	-0.9015	-0.2207	0.4663	-0.9015	0.1807	0.1807	0.4663

Ce individu passe directement a la génération suivante .

- **Mutation :**

Comme décrit précédemment, une mutation est effectuée en sélectionnant de façon aléatoire une position et en modifiant sa valeur. La mutation est effectuée en sélectionnant un seul parent en utilisant la méthode de sélection de Classe décrite ci-dessus. Pour l'application du projet, deux types des mutations sont effectuées:

- **Mutation** 1 : Cette mutation est appliquée à la probabilité de **10%** par position nucléotidique **PFM**. La figure **30** illustre un exemple. Les étapes de processus

de mutation sont les suivantes :

1. La position de nucléotide de la **PFM** est sélectionnée en utilisant une probabilité de 10% par position.
2. Pour la position choisie, la base (A, C, T ou G) est choisie au hasard.
3. La fréquence de la base est modifiée en fournissant une norme écart dans la plage de +/- 0,5.
4. Si la fréquence de la position est supérieure à 1, alors ces changements sont annulés et la valeur d'origine est restaurée.
5. Les fréquences des bases restantes sont encore choisie au hasard de telle sorte que la somme totale des fréquences de toutes les bases de la position sélectionnée ne dépasse pas 1.
6. Ajouter cette descendance à la nouvelle population

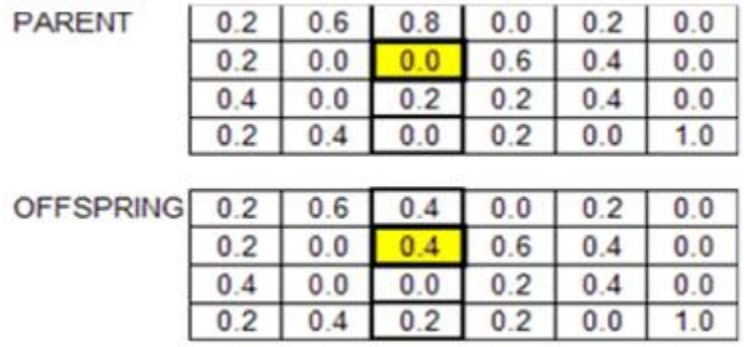

Figure 30 : Exemple de mutation

Ici, la cellule en jaune est mutée par un changement de fréquence de 0,4. Les cellules restantes de cette position sont ensuite modifiées de façon aléatoire.

Mutation 2 : Cette mutation est effectuée très rarement, avec une probabilité d'être appliquée seulement **4%** du temps. Elle consiste à ajouter une nouvelle colonne au hasard dans la PFM en cours d'examen. Cela augmente la taille du motif par1. La figure **31** montre un exemple de cette mutation.

Les étapes sont les suivantes :

1. Choisir au hasard la position à laquelle la nouvelle colonne doit être ajoutée.
2. Copiez toutes les valeurs de PFM en cours d'examen jusqu'à la position nombre généré de l'enfant
3. À la position sélectionnée, générer aléatoirement les fréquences pour les quatre bases. S'assurer que la somme des bases pour la position sélectionnée ne dépasse pas 1.
4. Copiez toutes les valeurs restantes de **PFM** origine dans la nouvelle génération.
5. Ajouter cette descendance à la nouvelle population

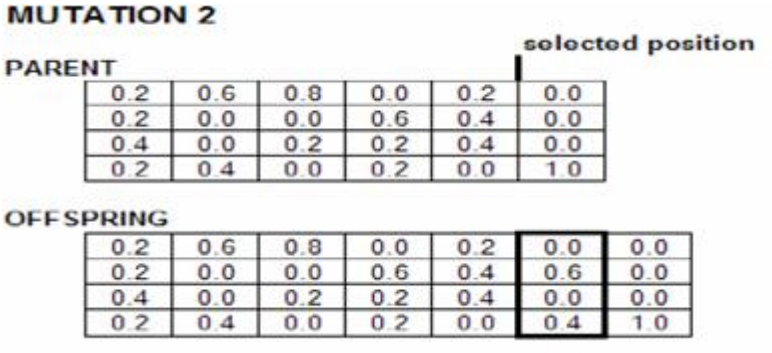

Figure 31 Exemple du processus de mutation2

- **Croisement:**

Dans des circonstances normales, les résultats de croisement dans la génération résultent en deux enfants. Le mécanisme de croisement utilisé dans le cadre de ce projet, toute fois génère une seule génération. La technique utilisée est appelée croisement uniforme. Dans le croisement uniforme, la valeur de chaque position de la génération est dérivée de un de ses parents normalement avec 50% de probabilité de sélection de un des deux parents.

Dans certains cas, le pourcentage de probabilité de sélectionner les données de position peut être différent, selon quels parents avec une probabilité plus élevée doivent

être favorisée La figure 32 montre un exemple de la façon dont la méthode de croisement uniforme travaille.

La méthode de croisement uniforme est appliquée comme suit:

1. Sélectionnez deux parents à l'aide de la méthode de sélection.
2. Prenez la taille de la progéniture égale à la taille du plus grand parent.
3. Pour chaque position de la progéniture, sélectionnez le parent dont les valeurs seront utilisées avec une probabilité de sélection de un des deux parents de 50%.
4. Copiez les valeurs de fréquence du parent sélectionné pour la position considéré.
5. Répétez les étapes (3) et (4) jusqu'à ce que le nombre des postions couverts = taille du plus petit parent.
6. Copiez les valeurs pour le reste des positions à partir du plus grand parent

   Ajouter cette descendance à la nouvelle population

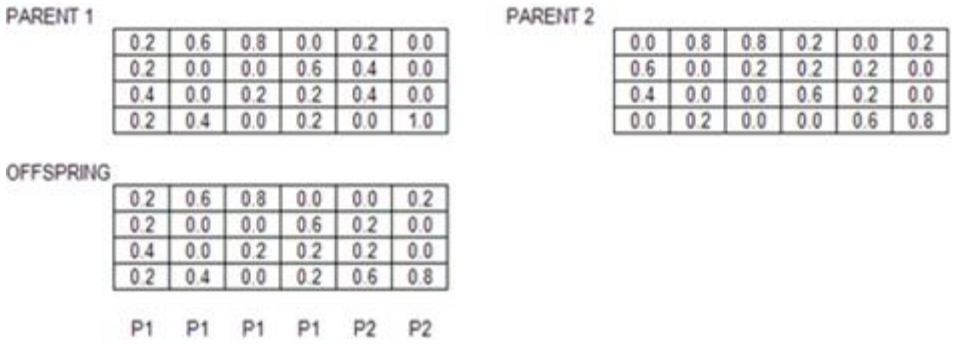

Figure 32 : Exemple de croisement Uniform

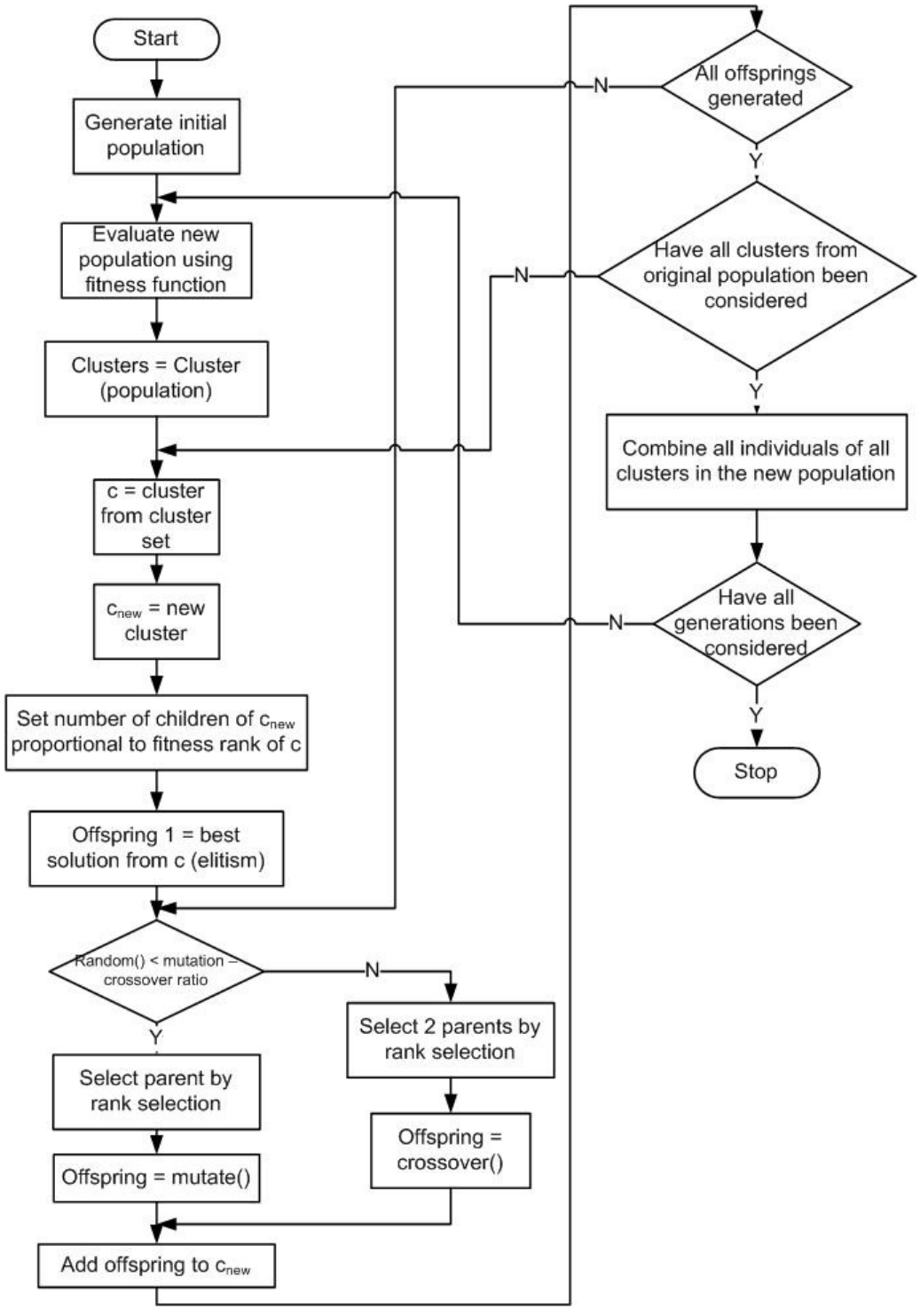

Figure 33 : Organigramme de l'algorithme génétique utilisé pour la découverte de motif

IV. Conclusion

Dans ce chapitre, nous avons essayé de expliquer en détaille l'algorithme génétique utilisé pour la Découverte des motifs et décrit en détail les différentes techniques utilisées pour l'implémentation de notre projet avec des exemples sur l'exécution des défirent fonctions.

Chapitre V : Implémentation et résultats expérimentaux

I. Introduction

Dans cette section, nous présentons le langage de programmation et les structures utilisées. Ensuite les aspects relatifs aux expérimentations effectuées et les résultats obtenus sont présentés. Des données synthétiques et réelles on été utilisées pour valider l'approche proposée. Pour plus de clarté nous commençons d'abord par la description des données dans les tests utilisées ensuite les résultats obtenus.

II. Introduction à Matlab

Pour l'implémentation de notre projet nous avons utilisé MATLAB. C'est un logiciel de calcul numérique, développé et commercialisé par la société américaine « The MathWorks ».

MATLAB (***matrix laboratory***) est un langage de programmation de quatrième génération, et un environnement interactif pour le calcul numérique, la visualisation et la programmation.

MATLAB permet de manipuler des matrices, d'afficher des courbes et des données, de mettre en œuvre des algorithmes, de créer des interfaces utilisateurs, et peut s'interfacer avec d'autres langages comme le C, C++, Java, et Fortran. (32)

III. Introduction à WebLogo :

Dans le cadre de ce projet, nous utilisons une représentation graphique de consensus générée par WebLogo.

Weblogo est une application Web pour la représentation des séquences biologiques. Un logo de séquence est une représentation graphique d'un acide aminé ou d'acide nucléique de l'alignement multiple de séquences développé par Tom Schneider et Mike Stephens. Chaque logo est constitué d'empilements de symboles, une pile pour chaque position dans la séquence. La hauteur totale de la pile indique que la conservation de la séquence à cette position, tandis que la hauteur des symboles à l'intérieur de la pile indique la fréquence relative de chaque aminoacide nucléique ou à cette position. En général, un logo de la séquence fournit une description plus riche et plus précise. (32)

IV. Un résumé sur les étapes de l'algorithme:

Considérons les ensembles de données d'entrée suivante :

Séquences de promoteur :

Séquence 1: AGTGCCGTGA

Séquence 2: GAGTGTGCAT

Séquence 3: TTTACGCTGG

Séquences arrière-plan :

Séquence 1: GAAACCAGCT

Séquence 2: ACATTTAGTA

Séquence 3: TTCGAATTGC

On commence par une population initiale **P** génère aléatoirement.

Par exemple en prend que la taille de la population initiale **P** égale a 5 (**PWM**)

Cette population représente la 1ère génération

Génération 1 = PWM1 , PWM2, PWM3 ,PWM4 ,PWM5

.

Pour chaque individu de la population en e calcule le fitness score a partir de la formule décrit précédemment.

Fitness score de PWM1 = **f1**

Fitness score de PWM2 = **f2**

Fitness score de PWM3 = **f3**

Fitness score de PWM4 = **f4**

Fitness score de PWM5 = **f5**

Apres en fait le regroupement de ces individus selon la méthode décrit précédemment.

En prend que le nombre des clusters égal à **2** et comme résulta que :

Le cluster **1** contient : **PWM1, PWM2, PWM3**

Le cluster **2** contient : **PWM4, PWM5**

Apres pour chaque cluster en applique les 3 opérations de reproduction :

Cluster 1

L'élitisme : l'individu qui contient la meilleur fitness score dans le cluster passe à la 2em génération en prend que f2 est supérieur que f1 et f3 . Alors **PWM2** passe à la 2em génération

Croisement et mutation : en appliquer ces deux opérations après la sélection des individus après le croisement ou la mutation les enfants génère est passe a la génération suivante .
Le croisement de PWM1 et PWM3 donne **PWM6**, en le met dans la deuxième génération

Génération 2 =PWM2, PWM6

De la même façon que le premier cluster en fait les 3 opérations de reproduction sue les restent des clusters

Dans le 2em cluster :

L'élitisme : l'individu qui contient la meilleur fitness score dans le cluster passe à la 2em génération en prend que **f5** est supérieur que **f4** . Alors **PWM5** passe à la 2em génération

Croisement et mutation : La mutation de PWM4 donne **PWM7**, en le met dans la deuxième génération

Génération 2 = PWM2, PWM6, PWM5, PWM7

En répète les opérations précédentes avec la génération 2 et ainsi de suite jusqu'à que le nombre des générations égale a le nombre des générations a atteindre

V. Description des données

Nous avons testé notre algorithme sur des séquences biologiques réelles et des séquences synthétiques (créé à partir de l'insertion des motifs dans des séquences réelles)

1. Données biologiques réelles

Les séquences biologiques réelles proviennent de l'homme, la souris, la levure, etc. Les séquences d'ADN qui contiennent des sites de fixation de facteur de transcription au niveau des régions promotrices définies expérimentalement et sauvegardées à la base de données **TRANSFAC.**

Les séquences d'arrière-plan représentant une partie aléatoire de l'ADN (réelles)

2. Données test synthétiques :

Nous avons également synthétisé des exemples de problème comme suit.

D'abord, un ensemble **N** de séquences de longueur **'L**'a été généré en sélectionnant aléatoirement les bases.

En second lieu, nous avons généré de la même manière un motif de longueur **'l**'.

En fin, nous injectons le motif généré dans des positions aléatoires. Ce processus nous permet d'avoir **N** séquences contenant le même et l'unique motif à des positions aléatoires.

VI. Les paramètres de L'algorithme

La table suivante présente les paramètres utilisés dans l'implémentation de l'algorithme génétique.

Paramètre	Valeur
La longueur de population initiale	**11– 15 n**
La taille de la population initiale	**20 PFMs**
Ratio (de mutation /croisement)	**7 : 3**
Ration de Mutation1	**100%**
Ration de Mutation2	**0%**
Fréquence de changement dans la Mutation1	**+0.5**
Probabilité de sélection pour les deux parents dans le croisement	**50%**
Nombre des clusters	**4**

Table 5: Table des paramètres

Pour l'exécution de notre programme, nous identifions les instances des paramètres de l'algorithme.

La première interface de l'application présente un ensemble de champs qui servent à introduire les paramètres utilisés. Après avoir rempli les champs avec les instances précédemment mentionnés, on démarre l'exécution en cliquant sur le bouton **Start.**

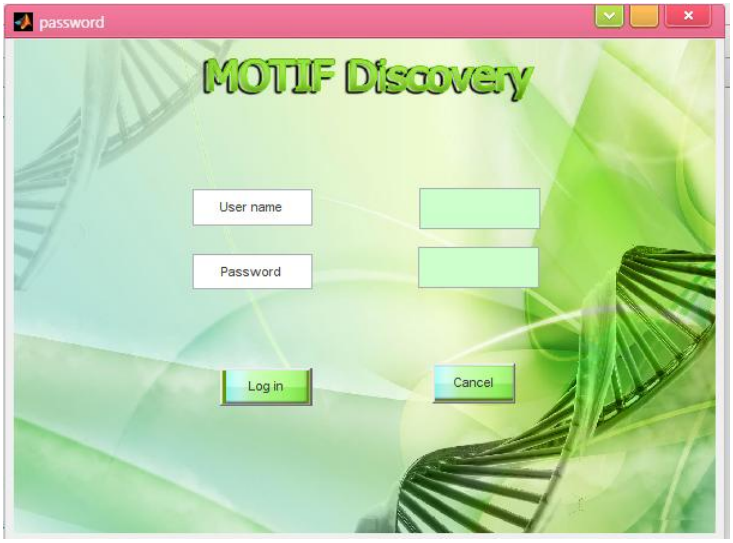

Figure 34: Interface d'authentification

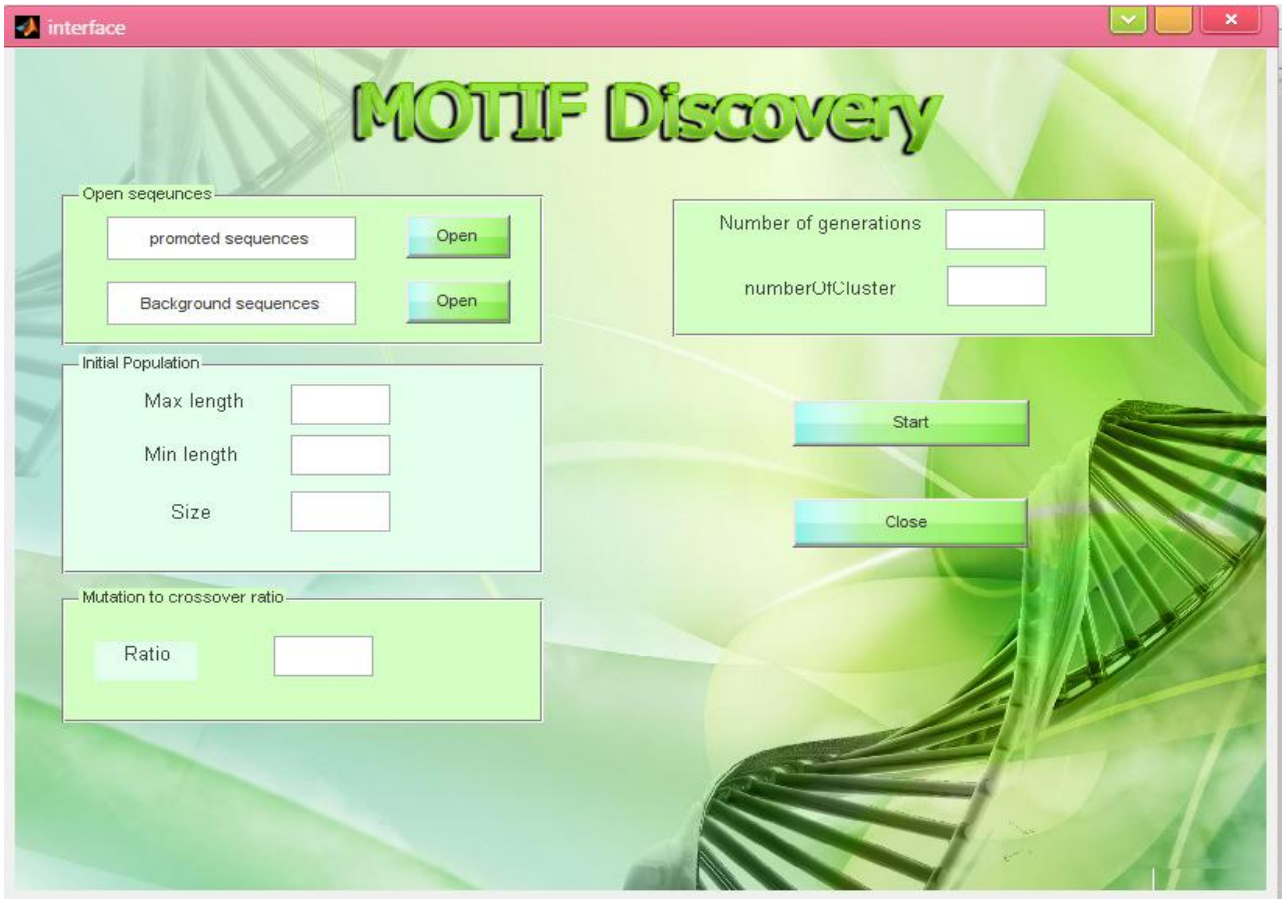

Figure 35 : Interface de programme

Premièrement, on sélectionne le fichier qui contient la séquence promoteur (le fichier doit être avec une extension **.txt** ou **.Fasta**) et le fichier qui contient la séquence d'arrière-plan

Après, nous ajoutons les paramètres :

- **Initial population :** représente la palette de la population initiale
- **Min length :** la taille minimale
- **Max length :** la taille maximale
- **Size :** nombre des matrices PFM dans la population initiale
- **Mutaion to crossover ration :** représente la proportion de mutation pour le croisement (cette valeur est proportionnelle à **10**)
- **Number of generations :** le nombre de générations
- **Number of cluster :** nombre de clusters.

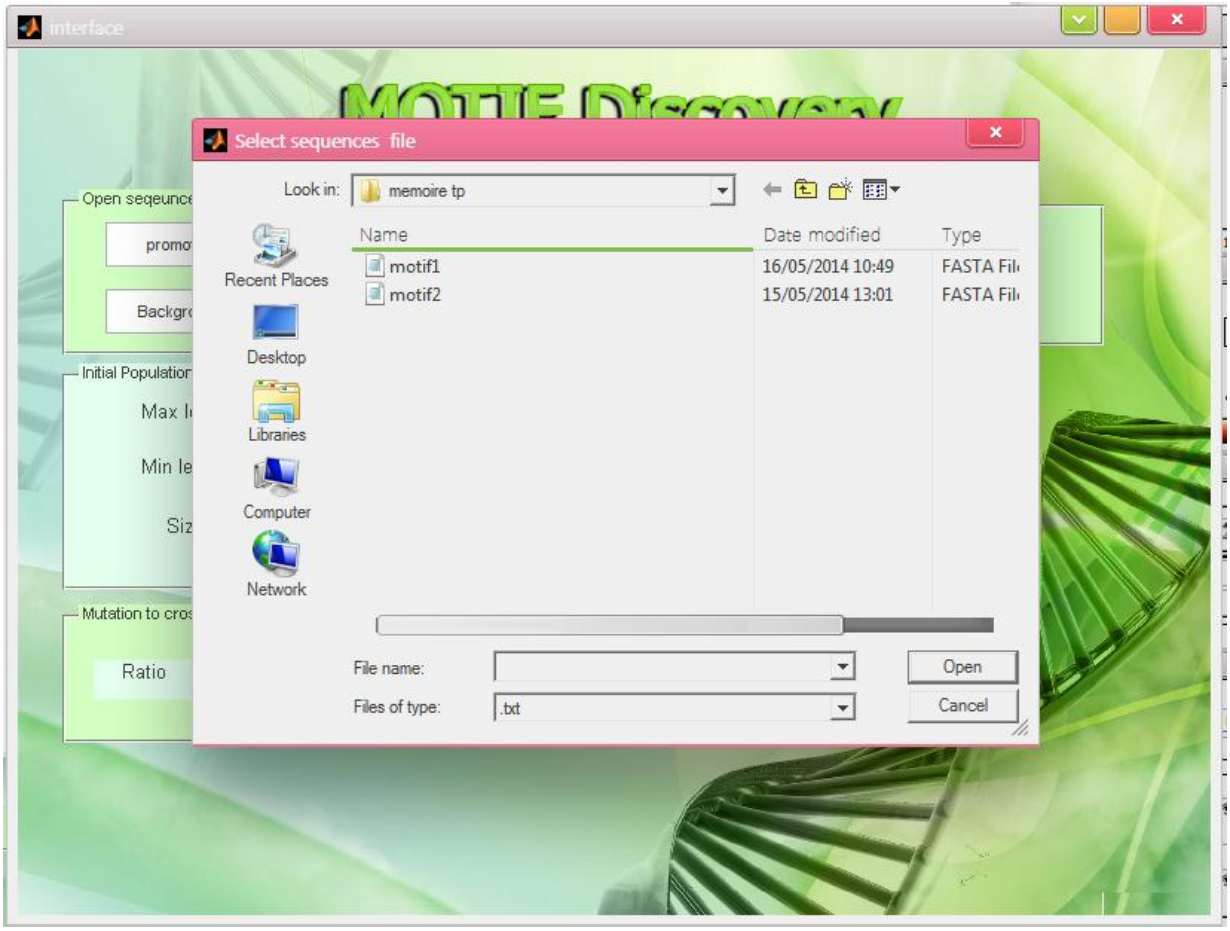

Figure 36 : Chargement des fichiers des séquences

Les valeurs des instances du test sont représentées dans la figure **(37)**

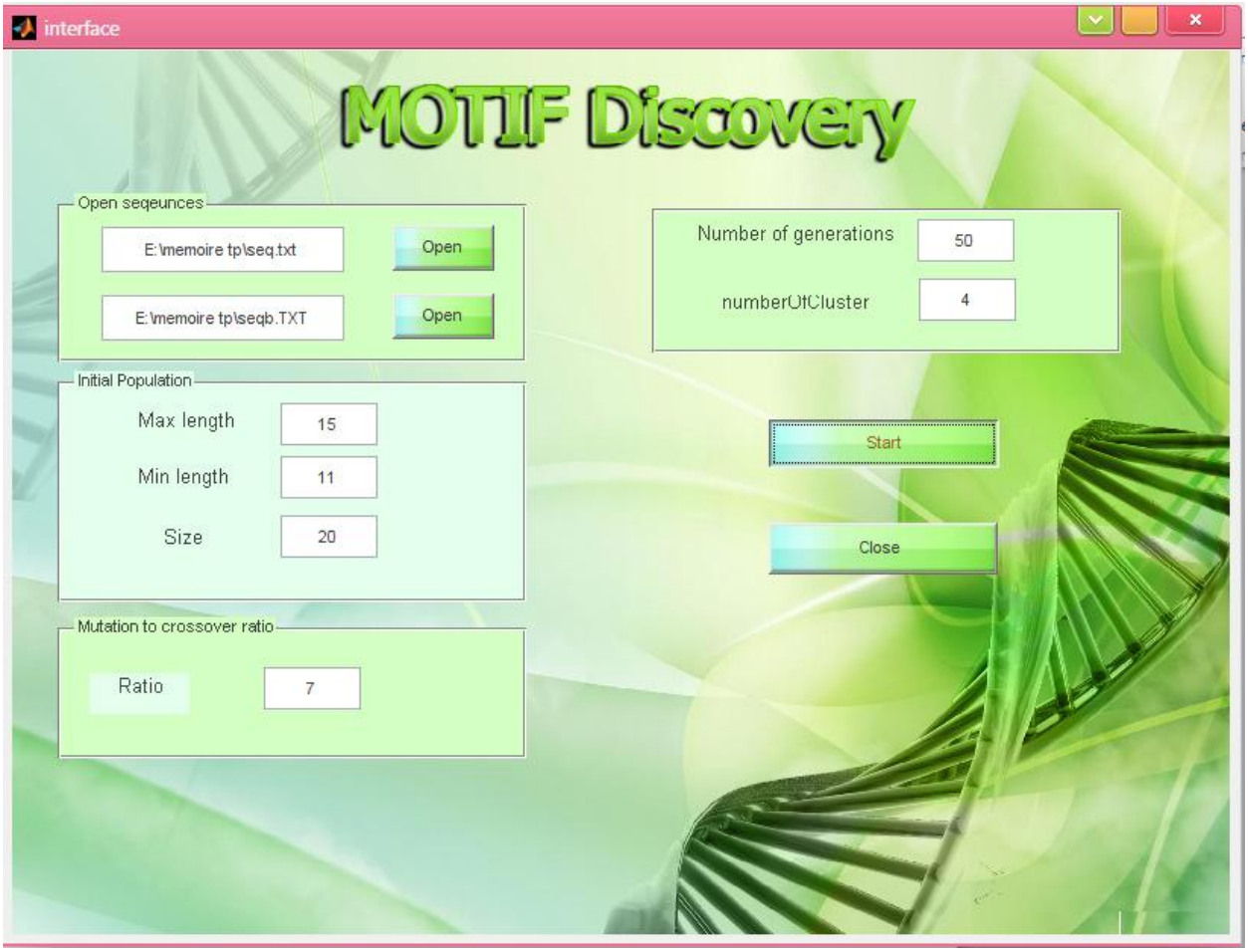

Figure 37 : Paramètres à introduire

VII. Tests et Résultats

3. Tests

Pour tester notre algorithme, On choisie un cas qui est le plus simple.

Dans ce test, on utilise une partie aléatoire d'ADN réel comme les séquences d'arrière-plan Apres, les motifs sont insérés dans plus que la moitié de ces séquences à des positions aléatoires.

En utilise un motif M1 :

```
motif1 =

TGAACATCAGAT
GAGCCAAACACA
AGACAAGTCTGG
AAGTCTGGTATT
TTAGGGTGTGGG
TTCATTATCTAT
GATCAGGATTAG
CCAAGGTCCACC
AAGGTCCACCAA
CTGTATTGGAGA


numberOFSeq =

    10


lengthOfseq =

    12
```

PFM de Motif 1 :

```
pfmMatrice =

    0.2995    0.3985    0.3985    0.2995    0.2995    0.2995    0.2005    0.2995    0.1015    0.3985    0.3985    0.2995
    0.2005    0.1015    0.1015    0.2995    0.2995    0.1015    0.1015    0.2005    0.4975    0.1015    0.2005    0.1015
    0.2005    0.2005    0.3985    0.2005    0.2005    0.2995    0.2995    0.2995    0.1015    0.2005    0.2995    0.2995
    0.2995    0.2995    0.1015    0.2005    0.2005    0.2995    0.3985    0.2005    0.2995    0.2995    0.1015    0.2995
```

Figure 38 : Motif consensus générer par WebLogo de motif1

Pour calculer la performance de cet algorithme, on l'exécute plusieurs fois (10 fois pour ce test) et on calcule le pourcentage des exécutions réussies.

Une exécution réussie de l'algorithme implique que la dernière solution évoluée doit être similaire à la séquence consensus du motif, la solution doit également avoir une longueur supérieure à 50% de la séquence consensus

On applique 10 essais et on calcule le pourcentage de succès du test.

Pourcentage de succès = nombre des essais réussie /nombre totale des essais

4. Les résultats

Dans cette section, nous présentons les résultats obtenus relatifs à la première génération et

une explication des étapes de l'algorithme. Aussi, un résumé sur les résultats de tous les tests qui ont été exécuté**s**.

- **Population initial :**

L'algorithme génère des matrices PFM aléatoirement de 20 individus avec une taille de 11 à 15 nucléotides

```
pop{1} =

    0.3993    0.3993    0.2500    0.1505    0.3495    0.1505    0.3495    0.2002    0.2002    0.4490    0.1007
    0.1505    0.1505    0.2500    0.1505    0.1007    0.3495    0.3993    0.2998    0.1007    0.1505    0.2998
    0.2500    0.1007    0.2500    0.3495    0.3993    0.2002    0.2002    0.2500    0.2500    0.2002    0.3993
    0.2002    0.3495    0.2500    0.3495    0.1505    0.2998    0.0510    0.2500    0.4490    0.2002    0.2002


pop{2} =

    0.1505    0.2500    0.3993    0.2002    0.2002    0.3495    0.0510    0.2998    0.3993    0.2002    0.2500    0.1505
    0.2998    0.3495    0.2500    0.2998    0.1505    0.2500    0.2998    0.2002    0.1007    0.3495    0.1505    0.3495
    0.2998    0.2002    0.2002    0.3495    0.2002    0.2500    0.3495    0.1505    0.2998    0.2002    0.2500    0.1007
    0.2500    0.2002    0.1505    0.1505    0.4490    0.1505    0.2998    0.3495    0.2002    0.2500    0.3495    0.3993


pop{3} =

    0.2500    0.2998    0.2998    0.3993    0.1007    0.2002    0.2500    0.3495    0.2998    0.2998    0.3495    0.2002    0.0510
    0.2500    0.1007    0.3993    0.1505    0.2998    0.2500    0.3495    0.1505    0.2002    0.1505    0.2002    0.2500    0.3495
    0.2500    0.2500    0.1505    0.2500    0.2998    0.3495    0.2002    0.2998    0.1505    0.3495    0.2500    0.1505    0.2998
    0.2500    0.3495    0.1505    0.2002    0.2998    0.2002    0.2002    0.2002    0.3495    0.2002    0.2002    0.3993    0.2998
```

Figure 39 : Des individus de la population initiale

- **Regroupement de population :**

L'algorithme exécute une fonction pour le regroupement de population. cette fonction calcule le fitness de chaque individu comme nous avons indiqué précédemment et retourne un vecteur qui contient le fitness de chacun**.**

```
fitscore =

  -0.9224  -0.6773  -0.6898  -0.6574  -0.7859  -0.9158  -0.6790  -0.8259  -0.6424  -0.8041  -0.8604  -0.7738  -0.7098  -0.7222  -0.8404  -0.9247  -0.7173  -0.7297  -0.7522  -0.8150
```

Et calcule la moyenne de ces valeurs = -0.7723

L'individu qui e**st** la plus proche valeur à la valeur moyenne devient le leader du premier cluster.

Le leader dans notre test est dans l'indice 12 (PFM12) qui a la valeur de fitness= -0.7738 En suivant, avec le calcul de la distance euclidienne entre chaque individu et le leader PFM12,

on trouve le leader du deuxième cluster. Ce leader a la plus grande distance.

Apres, on classe les individus dans le 1er cluster ou dans le 2eme cluster comme **suit**:

- On calcule la distance de chaque individu avec les deux leader**s**,
- On le classe avec le leader de valeur minimale
- On répète ce proce**ssus** jusqu'à obtention du nombre de cluster**s** égale à **4**

La fonction de cluster retourne un vecteur qui contient le numéro de cluster de chaque individu

```
clusterList =

     2     4     2     1     1     1     1     1     3     3     4     1     4     1     1     1     1     2     2     4
```

- ## Elitisme :

Pour chaque cluster, l'individu qui a la valeur maximale de fitness, la fonction d'Elitisme le choisi pour passer à la deuxième génération.

Exemple dans le cluster **numéro 1 : PFM4** à la valeur maximale, alors **PFM4** passe a la 2em génération.

- ## Mutation et croisement

Pour l'évaluation du cluster 1, L'algorithme choisi l'application de croisement

Apres, avec la fonction de roulette on choisi les deux parents pour faire le croisement

La roulette choisie PFM7 et PFM3

```
index1 =

     7


parent1 =

  Columns 1 through 13

    0.1505    0.0510    0.1505    0.2998    0.3495    0.1505    0.1505    0.2998    0.1007    0.2500    0.2998    0.3993    0.2998
    0.3495    0.2998    0.3495    0.2500    0.1007    0.3495    0.3993    0.3993    0.0510    0.2998    0.1007    0.1505    0.2002
    0.2500    0.1505    0.2500    0.1505    0.2998    0.2500    0.2002    0.1007    0.5485    0.2500    0.3993    0.2002    0.2500
    0.2500    0.4988    0.2500    0.2998    0.2500    0.2500    0.2500    0.2002    0.2998    0.2002    0.2002    0.2500    0.2500

  Columns 14 through 15

    0.2500    0.2002
    0.1505    0.2500
    0.2002    0.2500
    0.3993    0.2998
```

```
index2 =

     3


parent2 =

    0.2998    0.2998    0.2998    0.1505    0.1505    0.2500    0.2998    0.1505    0.2500    0.2500    0.2998    0.3495
    0.1007    0.1007    0.3495    0.3495    0.2998    0.2500    0.2002    0.2998    0.2500    0.3993    0.1007    0.2500
    0.1505    0.3495    0.1007    0.1007    0.3495    0.2002    0.2002    0.2998    0.2998    0.2500    0.1007    0.2002
    0.4490    0.2500    0.2500    0.3993    0.2002    0.2998    0.2998    0.2500    0.2002    0.1007    0.4988    0.2002


child2 =

  Columns 1 through 13

    0.1505    0.0510    0.1505    0.1505    0.1505    0.1505    0.2998    0.1505    0.2500    0.2500    0.2998    0.3495    0.2998
    0.3495    0.2998    0.3495    0.3495    0.2998    0.3495    0.2002    0.2998    0.2500    0.3993    0.1007    0.2500    0.2002
    0.2500    0.1505    0.2500    0.1007    0.3495    0.2500    0.2002    0.2998    0.2998    0.2500    0.3993    0.2002    0.2500
    0.2500    0.4988    0.2500    0.3993    0.2002    0.2500    0.2998    0.2500    0.2002    0.1007    0.2002    0.2002    0.2500

  Columns 14 through 15

    0.2500    0.2002
    0.1505    0.2500
    0.2002    0.2500
    0.3993    0.2998
```

Pour la 2^{em}évolution du 1er cluster, l'algorithme choisi l'opération de **Mutation** sur le parent indiqué dans la figure ci-dessous :

```
    parent =

    0.2002    0.3993    0.2002    0.2500    0.1505    0.3495    0.2998    0.2500    0.2998    0.1007    0.2500
    0.1505    0.3993    0.1505    0.3993    0.2002    0.2002    0.1505    0.1505    0.1505    0.2998    0.1505
    0.3993    0.1505    0.4490    0.2002    0.2998    0.1007    0.2002    0.4490    0.2998    0.2500    0.2500
    0.2500    0.0510    0.2002    0.1505    0.3495    0.3495    0.3495    0.1505    0.2500    0.3495    0.3495
```

Dans la 9 ^{eme} psition et le resulta dans la figure suivante :

```
child =

    0.2002    0.3993    0.2002    0.2500    0.1505    0.3495    0.2998    0.2500    0.7998    0.1007    0.2500
    0.1505    0.3993    0.1505    0.3993    0.2002    0.2002    0.1505    0.1505    0.1505    0.2998    0.1505
    0.3993    0.1505    0.4490    0.2002    0.2998    0.1007    0.2002    0.4490    0.2998    0.2500    0.2500
    0.2500    0.0510    0.2002    0.1505    0.3495    0.3495    0.3495    0.1505    0.2500    0.3495    0.3495
```

Le résultat final obtenu après toute les opérations dans la 1ère génération

```
resulta =

GTTCCCGCTTCCAG
GTTCCCGCTTCCCG
CTCTGCACGCGAATT
CTCTGCTGGCGAATT
GAGCTATGATT
GCGCTTTGATT
TACCAATATCTT
TTCCAATATCTT
TTGACAGTAGG
TTGACAGTTGG
CGATTTATAAGCA
CTTTTTATATGCA
TTTCGAGGCCCGAGA
TTTCGAGGCCCGCGT
TTTCGAGGCCCGGGA
TTTCGAGGCCCGTGA
AAAGGCCCTAG
AACTGCCCTAG
AAGGGCCCTAG
GGGAAGCCACACGC
GGGACGGGGCACGC
TGATCCTTACCAC
TGATGCTTAGCAC
TGATCCTTATCAC
ATCACCCATGATC
ATCAGCCATGATC
ATCATCCATGATC
TCAACTGTACTCTA
CCAGTAGTACTT
ATGAACTGAGATT
```

On applique l'algorithme avec 50 générations et avec 100 générations

Le tableau suivant représente un résumé de les résultats obtenu :

Motif	La longueur de séquence	Nombre d'essais	Nombre des générations	Nombre des séquences promoteur	Nombre des séquences d'arrière-plan	Pourcentage de succès
Motif1	**100**	**10**	**50**	**50**	**50**	**5/10 = 50%**
Motif1	**100**	**10**	**100**	**50**	**50**	**13/15 = 87%**

Table 6 : Les résultats des tests

Les résultats pour 50 générations donnent 5 essais avec succès d'un totale de 10 essais Mais pour 100 générations en obtient 13 essais avec succès de trouver le motif de totale des 10 essais.

- ✓ En général, nous avons observé que la fitness du motif évolue et augmente lorsque le nombre de générations augmente.
- ✓ L'expérience avec 50 générations ne donne pas une très bonne solution. L'expérience avec 100 générations donne une meilleure solution.
- ✓ Le pourcentage de succès est doublé avec l'augmentation du nombre de générations.

- **Le temps d'exécution :**

Pour cet algorithme, Nous constatons que la durée moyenne de l'exécution d'une seule génération avec les instances montré dans la table précédente est de : 42 seconds.
Pour 50 générations l'logarithme prend un temps plus long qui est de : 35 minute.

VIII. Conclusion

Dans ce chapitre, nous avons décrit brièvement l'environnement de développement de l'algorithme, ensuit nous avons fait des tests sur notre algorithme et décrit les résultats obtenus. Aussi, nous avons expliqué les phases suivies par notre programme pour la résolution de ces tests.

Conclusion Générale et perspectives

5. Conclusion Générale

Dans le cadre de ce travail de master, nous avons traité un problème très important en bioinformatique celui de la découverte de motif. Ce problème dans ce mémoire été défini comme un problème mono-objectif où les différentes méthodes développées cherchent à optimiser une seule fonction objectif.

Avant le développement de notre approche nous étions obligés de comprendre quelques notions de la biologie moléculaire et faire une brève étude sur les méthodes proposées qui nous ont aidés de bien comprendre et formuler notre problème.

Ensuite, nous avons présentées le problème de découverte de motif commun à un ensemble de séquences biologiques.

Nous avons proposée est une méthode d'optimisation basée population en choisissant les algorithmes génétiques.

Les résultats expérimentaux obtenus montrent que les AG sont efficaces pour ce problème et surtout lors la combinaison de l'AG avec le clustering qui aide à déterminer la capacité à identifier de meilleures solutions.

Une série d'expériences ont été effectuées sur divers ensembles de données pour tester si l'algorithme a atteint son objectif d'identifier les motifs présents dans les séquences biologique. Les résultats montrent que cette approche fonctionne très bien lors de l'identification d'un seul Motif de l'ensemble de la séquence de promoteur.

Perspectives

Comme perspectives, nous souhaitons de raffiner l'approche dans le sens de maintenir la diversité, en évitant une convergence prématurée. Une validation plus approfondie sur des données réelles est planifiée. En revanche, Comparer la performance de notre approche avec d'autres méthodes de découverte de motifs comme MEME, AlignACE, etc. est envisageable.

Après que cette approche prouve son efficacité à résoudre le problème de découverte de motif sous l'ongle mono-objective, nous espérons le traiter sous l'angle de l'optimisation multi-objective.

Bibliographie

1. **Quinkal, Isabelle.** *INRIA Rhône-Alpes.* Septembre 2003.

2. **TOURS, Université de.** *GÉNET.* 2006.

3. **Talla, Emmanuel.** *Cours de Bioinformatique Appliquée (Partie 2).* Marseille : s.n., 2008-2009.

4. **Bonsai, Equipe.** *Cours d'introduction à la bioinformatique et de présentation des banques de séquences (1er partie).* 2012.

5. **Thieffry, Denis.** *De la Bioinformatique.* Marseille, France : Université de la Méditerranée.

6. **Ghosh, Status of the transcription factors database (TFD). Nucl. Acids Res. (1993) 21(13): 3117-.**

7. **E. Wngender, P. Dietze, H. Karas et R. Knüppel, TRANSFAC: a database on transcription factors.**

8. **R. Knuppel, P. Dietze, W. Lehnberg, K. French et E. Wingender, TRANSFAC retrieval program: a.**

9. **Zeltni, Kamel.** *Découverte de Motifs Multicritères par Utilisation de Métaheuristique à Comportement Quantique.*

10. **C. Kanz, P. Aldebert, N. Althorpe, W., A. Baldwin, K. Bates et all.** *The EMBL Nucleotide Sequence, Nucleic Acids Research.* 2005.

11. **D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell et D. L. Wheeler.** *GenBank, Nucleic Acids Research.* 2007.

12. **Apweiler, Bairoch et R.** *The SWISS-PROT protein sequence database and its supplement,Nucleic Acids Research.* 2000.

13. **H. M. Berman, J. Westbrook, Z. Freng, G. Gilliland, T.Bhat, H. Weissig, I. Shindyalov et P. Bourne.** *The protein data bank, logo.*

14. **web.** Qu'est-ce que la pharmacogénétique? Qu'est-ce que la pharmacogénomique. [En ligne] 2005. http://omics-ethics.org/fr/definition-pharmacogenomique.

15. **wikipedia.** Génomique comparative. *wikipedia .* [En ligne] http://fr.wikipedia.org/wiki/G%C3%A9nomique_comparative.

16. **HAMIDECHI, DJEKOUN A. Dr.** *Cours de bioinformatique : Les Motifs Nucléiques et Protéiques.* s.l. : Université Mentouri-Constantine Faculté des Sciences de la Nature et de la Vie.

17. **Benlahrache, Nadira.** *Optimisation Multi-Objectif Pour l'Alignement.*

18. **THIEFFRY, Denis.** *Découverte et recherche de motifs dans les séquences de macromolécules*

biologiques.

19. **bioinformatique.** *Bioinformatique et modélisation Finding motifs in sequence Pattern matching.*

20. **MANCHERON, Alban.** *Extraction de Motifs Communs dans un Ensemble de Séquences. Application à l'identification de sites de liaison.* 2006.

21. HiddenMarkovModel. [En ligne] http://www-igm.univ-mlv.fr/~dr/XPOSE2012/HiddenMarkovModel/exemples.html.

22. **TERRAPON, Nicolas.** *Recherche de domaines protéiques divergents à l'aide de modèles de Markov cachés : application à Plasmodium falciparum.*

23. **Sumazin, G. Chen, N. Hata, D. A. Smith, T. Zhang et M. Q. Zhang.** *DWE: discriminating word enumerator, Bioinformatics vol. 21, no. 1.* 2005.

24. **Sagot, M.** *Spelling approximate repeated or common motifs using a suffix tree, Lecture Note in Computer Science.* 1998 .

25. **J. Vilo, A. Brazma, I. Jonassen, A. Robinson et E. Ukonnen.** *Mining for putative regulatory elements in the yeast genome using gene expression data, In Proceeding of the Eighth .* 2000.

26. **Stephens, T. Schneider et R.** *Sequence logos: a new way to display consensus sequences, Nucleic.* 1990.

27. **Siarry, Y. Collette et P.** *Optimisation Multiobjectif.* s.l. : ÉDITIONS EYROLLES, 2002.

28. **MAGNIN, Vincent.** Optimisation et algorithmes génétiques. [En ligne] 2006. http://magnin.plil.net/spip.php?article47.

29. **Meshoul, S.** *Recherche Opérationelle Avancée.* Algerie : s.n., 2011.

30. **MAHDI, SAMIR.** *Optimisation Multiobjectif Par Un Nouveau Schéma De.*

31. **Ismail, S. Ben.** *Introduction à l'optimisation combinatoire.* 2012.

32. **Berro, Alain.** *Optimisation multiobjectif et statégies d'evolution en environnement dynamique .*

33. **Pierre, CHAPTAL Thomas – ESPIEUX.** *Algorithmes génétiques.* 2012.

34. algorithmes génétiques. *SIS - Université du Sud .* [En ligne] http://sis.univ-tln.fr/~tollari/TER/AlgoGen1/node5.html.

35. **RadetFrancois-Gérard, Souquet Amédée et.** *ALGORITHMES GENETIQUES.*

36. wikipedia. *wikipedia.* [En ligne] http://fr.wikipedia.org/wiki/MATLAB.

37. weblogo. *weblogo.* [En ligne] http://weblogo.berkeley.edu/.

38. algorithmes génétiques. [En ligne] http://sis.univ-tln.fr/~tollari/TER/AlgoGen1/node5.html.