ا لجمهورية الجزائرية الديمقراطية الشعبية
RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي و البحث العلمي
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

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

جامعة الإخوة منتوري قسنطينة
كلية علوم الطبيعة و الحياة

Département : Microbiologie

قسم : الميكروبيولوجيا

Mémoire présenté en vue de l'obtention du Diplôme de Master

Domaine : Sciences de la Nature et de la Vie

Filière : Biotechnologie

Spécialité : *Mycologie et biotechnologie fongique.*

Intitulé :

Modélisation du processus de la traduction d'une séquence d'ADN naturelle et d'une séquence chimère en séquence protéique

Présenté et soutenu par : *Chaabani Aya*
Douadi Khaoula

Le : 14/07/2019

Jury d'évaluation :

Présidente du jury : *Mme. Abdelaziz Ouidad* (Maitre de conférences B- UFM Constantine).

Rapporteur : *Mme. Djama Ouahiba* (Maitre assistante A- UFM Constantine).

Dr. Arabet Dallel (Maitre de conférences A- UFM Constantine).

Examinateur : *Dr. Chehili Hamza* (Maitre de conférences B- UFM Constantine).

Année universitaire
2018 - 2019

Remerciements

D'abord nous tenons à remercier en premier lieu Dieu le tout puissant,

De nous avoir donné la force et le courage afin de réaliser cette étude.

Nous remercions vivement et chaleureusement notre encadreuses : ***Mme Djama Ouahiba*** *et* ***Dr. Arabet Dallel****, pour avoir encadrées et dirigées ce mémoire avec une grande rigueur scientifique.*

On les remercier particulièrement ses disponibilités et ses conseilsjudicieux, ainsi que ses patiences qui ont contribué à la réalisation et l'accomplissement de ce travail.

C'est avec un très grand plaisir nous remercions ***Mme. Abdelaziz Ouidad****, Maitre de conférences B à la faculté des sciences de la nature et de la vie, l'université des frères Mentouri Constantine, pour avoir accepté de présider le jury de notre soutenance.*

C'est avec un très grand plaisir nous remercions ***Dr. Chehili Hamza,*** *Maitre de conférences B à la faculté des sciences de la nature et de la vie, l'université des frères Mentouri Constantine, pour avoir accepté de présider le jury de notre soutenance, pour avoir accepté de présider le jury de notre soutenance.*

Nous souhaitons exprimer égalementmes remerciementsainsi que mes profondes gratitudes à chefde département de Microbiologie ***Mr. Farhati laid.***

Dédicace

Dieu tout puissant merci d'être toujours auprès de moi

- ✓ *A l'homme de ma vie, mon exemple éternel, mon soutien moral, rien au monde ne vaut les efforts, fournis jours et nuits pour mon éducation et mon bien être, j'espère avoir répondu aux espoirs que tu as fondé en moi. que Dieu tout puissent te garde santé, bonheur et longue vie.*
  Mon très cher papa HOCINE.

- ✓ *A la lumière de mes jours, la source de mes efforts, la flamme de mon cœur, ma vie…Aucune parole pour exprimer mon amour et mon attachement à toi. Merci tu m'as toujours donné de ton temps, de ton énergie, de ton cœur et de ton amour. Que Dieu tout puissent te garde santé, bonheur et longue vie.*
  Ma très chère mère NADIA.

- ✓ *A qui ma toujours aidée, écouté et encouragé tout au long de mon parcours ;ma charmante, ma meilleure, ma unique, ma chère sœur* ***HAWA*** *l'amour et la fraternité nous unissent à jamais. Que Dieu te garde pour moi.*
- ✓ *A mon petit, mon adorable, ma joie, ma raison de vivre ; le meilleur frère qui existe* ***MOUAD.*** *Je te souhaite un avenir plein de joie et de réussite.*

Je dédie ce modeste travail :

A toutes mes amies et surtout mes chères ***YOUSRA., Maroua, Meissa, Mira, Fatima, Rima.***

A mon cher binôme ***KHAOULA.***

A tout mes collègues de promotion.

A tout mes enseignions de biologie et surtout de mycologie.

Finalement Très sincèrement et du plus profond du cœur je remercie

Tout ceux que j'aime, tous ce qui sont proche de mon cœur et d'ont je n'ai pas cité le nom

Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect

Que j'ai toujours eu pour vous. Que Dieu vous protège pour moi.

Aya

Dédicace

Dieu tout puissant merci d'être toujours auprès de moi

Je dédie ce travail :

A ma chère mère,

A mon cher père,

Qui n'ont jamais cessé de formuler des prières à mon égard de me soutenir et de m'épauler pour que je puisse atteindre mes objectifs.

A mes frères, Chouaib, Moussaab ; Billel

A mes chères soueurs Sara et Rima

A ma famille

A mon cher binôme Aya

Qui m'a aidé et supporté dans les moments difficiles

A mes chères amies Sara Rayene Roumeissa Zienb

A tous ceux que j'aime et ceux qui m'aiment

Khaoula

Table des matières

PARITIE PRATIQUE

chapitre 3 : Matériels et méthodes

Chapitre 4 : Résultats et discussions

Listes

Des figures, tableaux et abréviations

Liste des figures

.

Liste des tableaux :

Liste des abréviations

ADN: Acide désoxyribonucléique.

ARN: Acide ribonucléique.

ARNm: Acide ribonucléique messager.

ARNt: Acide ribonucléique de transfert.

IF: *Initiation Factor.*

EF: *Elongation Factors.*

RF: *Release Factors.*

eIF: *Eukaryotic Initiation Factors.*

eEF: *Eukaryotic Elongation Factors.*

eRF: *Eukaryotic Elongation Factors.*

CDS: *Coding Sequence.*

BLAST: Local Alignment Search Tool.

NCBI: *National Center For Biotechnology Information.*

MATLAB: *Matrix Laboratory.*

EMBL: *European Molecular Biology Laboratory.*

U : Uracile.

T : Thymine.

C : Cytosine.

A : Adénine.

GTP : Guanosine triphosphate.

Résumé :

Ce travail a été réalisé dans le but de développer un logiciel de modélisation de l'information génétique en protéines. Nous avons pu mettre au point un logiciel qui a la capacité de lire des séquences d'ADN qu'elles soient réelles (qui existent dans la nature) ou chimères (imaginées). Cette modélisation a été implémentée dans le langage MATLAB. Le logiciel a été par la suite vérifié et validé. D'après les résultats, on peut dire que notre logiciel possède la capacité de simuler le processus naturel de la traduction des séquences d'ADN en protéines. Ainsi, le logiciel développé est capable de traduire des séquences d'ADN chimère en protéines et peut ainsi servir dans les recherches de plusieurs domaines tels que le domaine : pharmaceutique, cosmétique et industrie.

Les mots clés : ADN ; ARN ; Protéines ; Acide aminés ; Séquence chimère ; Programme ; Modélisation ; Simulation ; Alignement.

Abstract:

This work was done with the aim of developing a soft ware for modelling genetic information in proteins. We have been able to develop software that has the ability to read DNA sequences that are real (that exist in nature) or chimera (imagined).This modelling was implemented in the MATLAB language. The software was subsequently verified and validated.From the results, we can say that our software has the ability to simulate the natural process of translating DNA sequences into proteins.Thus, the software developed is capable of translating chimeric DNA sequences into proteins and can thus be used in the research of several fields such as the field: pharmaceutical, cosmetic and industry.

Keywords: DNA; RNA; Protein; Amino acid; Sequence chimera; Program; Modeling; Simulation; Alignment.

ملخص:

تم هذا العمل بهدف تطوير برنامج لتصميم المعلومات الوراثية في البروتينات. لقد تمكنا من تطوير برنامج لديه القدرة على قراءة تسلسل الحمض النووي الحقيقي (الموجود في الطبيعة) أو الوهمي (المتخيل). تم تنفيذ هذا النموذج باستعمال لغة البرمجة MATLAB . أيضا قمنا بالتحقق من البرنامج والتثبت من صحته. من خلال النتائج،يمكننا القول أن برنامجنا لديه القدرة على محاكاة العملية الطبيعية لترجمة تسلسلا لحمض النووي إلى بروتينات. وبالتالي، فإن البرنامج الذي تم تطويره قادر على ترجمة تسلسل الحمض النووي الوهمي إلى بروتينات، وبالتالي يمكن استخدامه في البحث في العديد من المجالات مثل مجال: الأدوية ومستحضرات التجميل والصناعة.

الكلمات المفتاحية: حمض نووي صبغي, حمض نووي ريبوزي, البروتينات, الأحماض الأمينية, تسلسل خيالي, البرنامج, النمذجة, للمحاكاة, الانحياز.

Introduction

Au cours des années, les biologistes ont collecté une grande masse d'informations concernant les séquences nucléotidiques et protéiques. Cependant, le traitement et l'interprétation du contenu de ses séquences est devenus très pénibles.

Pour cela, l'introduction d'outils puissants et de méthodes efficaces pour l'analyse et l'interprétation des données biologiques sont primordiales. Partant de ce besoin, la bioinformatique en tant que domaine scientifique a émergé.

La bioinformatique est une discipline relativement récente qui est actuellement en plein essor que ce soit en biochimie, biologie structurelle, moléculaire, immunologie et tous les domaines de la recherche (Layeb, 2005).

La bioinformatique des protéines permet de développer de manière rationnelle les expériences nécessaires à l'étude efficace des protéines comme toute approche scientifique.

La modélisation et la simulation informatique permettent d'exprimer à la machine (L'ordinateur) le fonctionnement d'un processus naturel d'une part et d'améliorer les études sur les processus naturels par l'être humain d'autre part. Parmi les processus naturels qui peuvent être modélisés pour créer des simulations informatiques le processus de la synthèse des protéines à partir d'une séquence d'ADN d'un gène.

C'est pourquoi nous posons la question suivante : comment on peut faire la modélisation informatique du processus de la traduction des séquences d'ADN en séquences protéiques ? (Mehdi et Meziani, 2018).

L'objectif de cette étude est la modélisation informatique d'un processus naturel qui est la traduction des séquences d'ADN en séquence d'acides aminés. Cette modélisation permet de maitriser la complexité et d'abstraire la réalité pour mieux comprendre le système à réaliser. Elle permet de générer virtuellement une séquence d'acides aminés à partir d'une séquence d'ADN naturelle (qui existe dans la nature).

Aussi, nous avons modélisé une séquence d'ADN chimère et on a obtenu une séquence protéique qui peut être n'existe pas dans la nature. Cette modélisation nous permet de construire des nouvelles séquences protéiques qui ont un rôle très important à l'industrie pharmaceutique et médicale comme par exemple la construction d'une nouvelle enzyme, un nouveau médicament…

Ce mémoire est organisé en 4 chapitres :

D'abord, le premier chapitre est une explication du processus naturel qui est la traduction des séquences d'ADN en protéines.

Le deuxième chapitre présente des notions informatiques et bioinformatiques qui entrent dans la modélisation d'un processus naturel.

Le troisième chapitre permet de mettre en évidence la modélisation informatique de processus de la synthèse des protéines à partir d'ADN en utilisant le langage MATLAB.

Finalement, le dernier chapitre est consacré à présenter les résultats et les discutions.

Le manuscrit s'achève par une conclusion et des perspectives.

Chapitre : 01

Notions sur l'ADN

1. Histoire de la découverte de l'ADN

Au début des années 50 Watson et Crick ont découvert que l'ADN est la molécule portant l'information génétique Elle est compactée dans des structures particulières appelées chromosomes.

Les biologistes de l'époque ont compris que la molécule de l'ADN se compose de quatre types de base azotée : A(Adénine), T(Thymine), C(Cytosine), G(Guanine) Qui forment en association avec un sucre (le désoxyribose) et l'acide phosphorique ce que l'on appelle les nucléotides

Ces bases azotées sont complémentaires grâce à des liaisons hydrogènes : A avec T, C avec G.

Les recherches ont montré que le positionnement des nucléotides par complémentarité donne à la molécule d'ADN une structure hélicoïdale en forme de double hélice.

Aujourd'hui, plus de cinquante ans après la découverte du double hélice, cette description initiale reste vraie et n'a pas été modifiée par les nouvelles découvertes (Watson et Crick, 2012).

2. Définition de l'acide désoxyribonucléique

L'acide désoxyribonucléique est un enchainement de nucléotides Il se compose de deux chaines antiparallèles où les bases azotées sont liée entre elles par des liaisons hydrogènes tournées vers l'intérieur tandis que le désoxyribose et les acides phosphoriques sont tournée vers l'extérieur (Housset et Raisonnier, 2009).

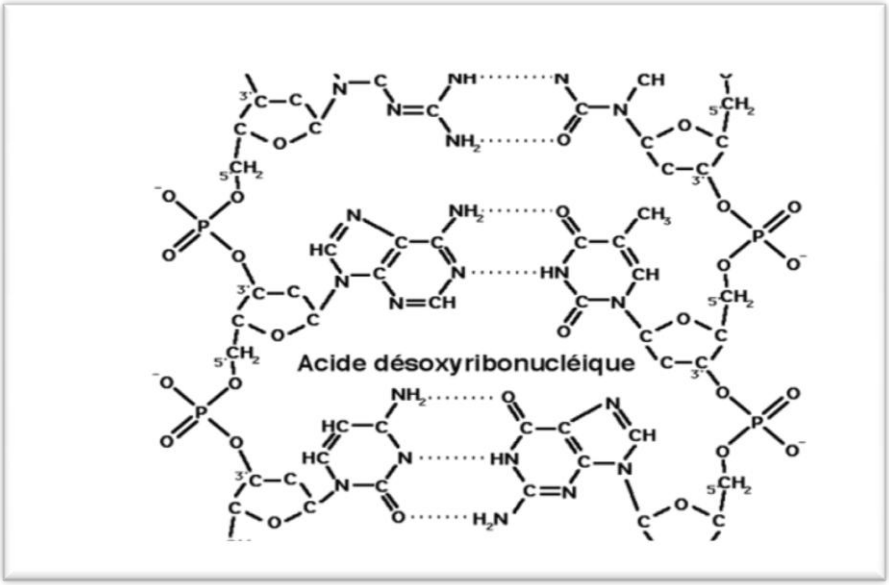


Figure 1. Structure de l'acide désoxyribonucléique (Housset et Raisonnier, 2009).

3. La structure de l'ADN

3.1. Les bases

L'acide désoxyribonucléique contient 4 bases qui sont A, T, C et G. Il y a 2 catégories de bases :

- Les purines : constituées de deux cycles aromatiques.
- Les pyrimidines : constituées d'un seul cycle aromatique.

Les atomes de carbone et d'azote des cycles aromatiques sont numérotés de 1 à 9 (base puriques) et de 1 à 5 (bases pyrimidiques) (Luchetta, 2009).

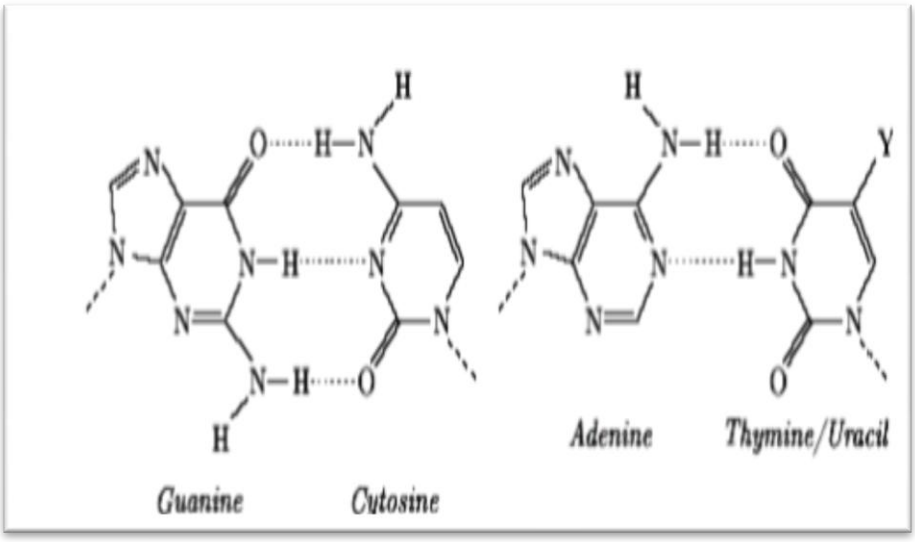


Figure 2. Schéma des bases puriques et pyrimidiques (Meshoul, 2005).

3.2. Le désoxyribose

C'est un pentose c'est-à-dire un ose qui se compose de 5 carbones. Il dérive du ribose par une réduction de la fonction alcool du carbone n°2 (Luchetta ,2009).

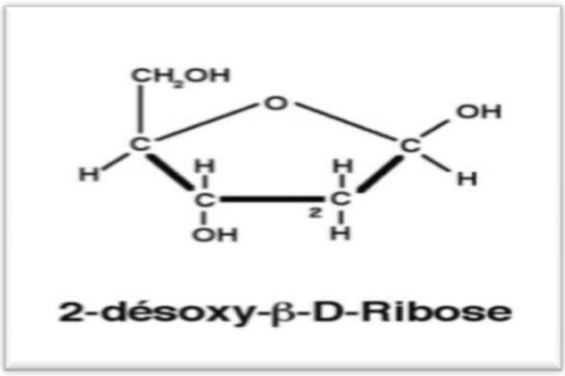


Figure 3. Formule en perspective du désoxyribose (Housset et Raisonnier, 2009).

3.3. L'acide phosphorique

Donne un groupement phosphate.

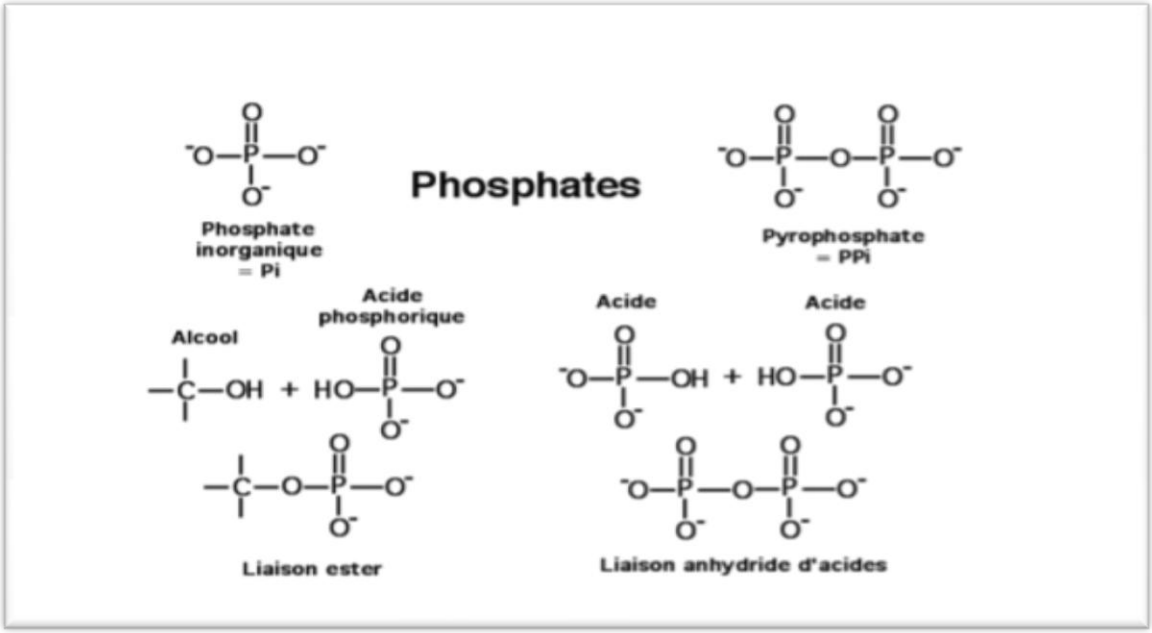


Figure 4. Structure de l'acide phosphorique (Housset et Raisonnier, 2009).

3.4. Le nucléoside

C'est le résultat de l'association d'une base azotée avec un sucre, donc le nucléoside se compose d'une base et d'un sucre liée par liaison N-osidique.

Il est très important de prendre en considération la numérotation dans un nucléoside, donc les atomes de base par des chiffres : 1, 2, 3, ect et pour les distinguer les carbones du sucre sont numérotés : 1',2',3'…..etc. (Housset et Raisonnier, 2009).

3.5. Le nucléotide

La formation de nucléotides fait intervenir l'association d'une base azotée liée par une liaison osidique avec un sucre. Ce dernier est lié d'autre part par une liaison ester à un phosphate (Housset et Raisonnier ,2009).

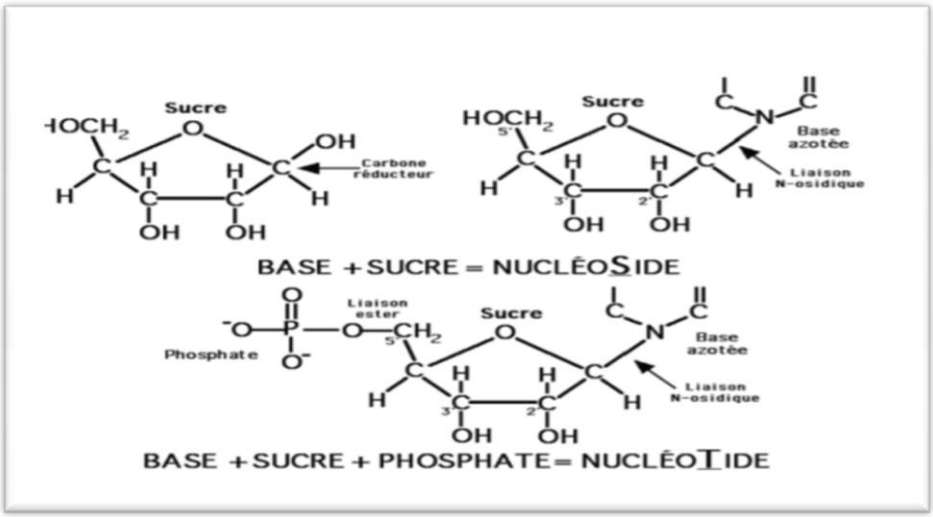


Figure 5. Nucléoside et Nucléotide (Housset et Raisonnier, 2009).

Les atomes de carbones du cycle de la base azotée sont numérotés 1, 2, 3, etc., et ceux du sucre 1',2',3', etc.

4. La double hélice

La structure de la double hélice est un enroulement de deux brins, associés entre eux par des liaisons hydrogènes (Adénine avec la thymine par deux liaisons hydrogènes ou guanine cytosine par 3 liaisons hydrogènes).La formation de la double hélice engendre la formation d'un grand et d'un petit sillon (Boujard *et al.*, 2012).

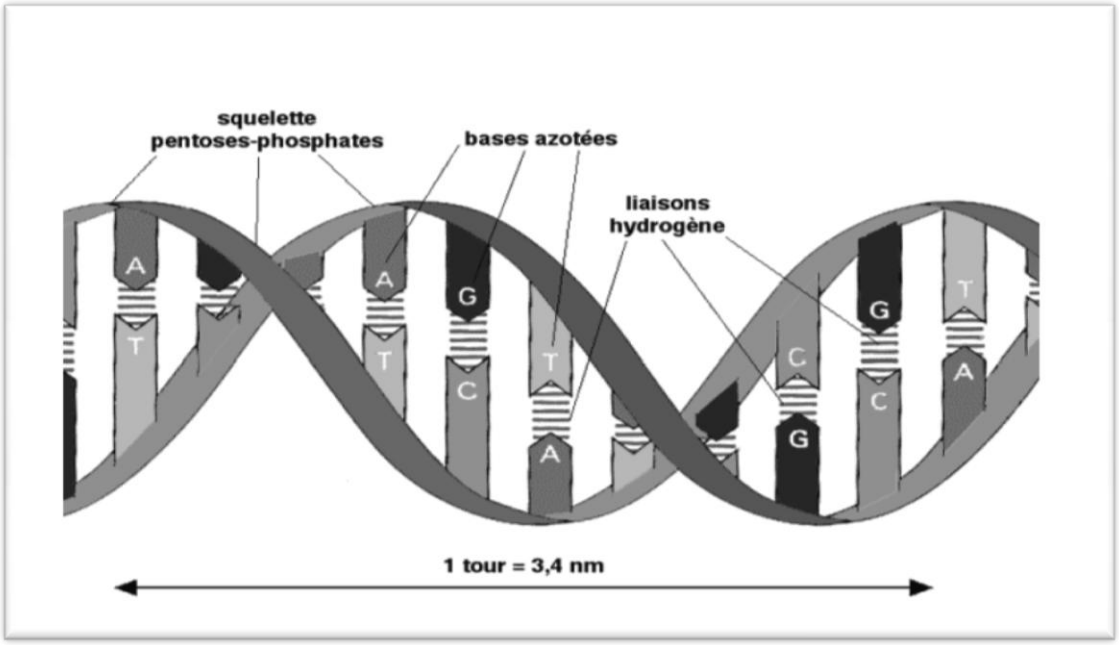


Figure 6. Illustration schématique qui représente l'enroulement des deux brins d'ADN (Housset et Raisonnier, 2009).

A partir de la séquence d'ADN, on peut obtenir une séquence protéiques, donc la biosynthèse des protéines est l'ensemble des processus biochimiques permettant aux cellules de produire leurs protéines à partir de leurs gènes, elle recouvre les étapes de transcription de l'ADN en ARNm d'aminoacylation des ARN de transfert, de traduction de l'ARN messager en chaines polypeptidiques**.**

5. La transcription

La transcription c'est la transformation de l'information génétique contenue dans l'ADN en une copie de structure différente mais portant fidèlement la même information. Il s'agit de la formation de l'ARNm.

La transcription est un processus hautement régulé. Il résulte de la fixation de facteurs de transcription sur les séquences d'ADN situées à proximité du gène à régulé pour assurer la transmission intégrale et fidèle de l'information porté sur le gène.

Enfin l'ARNm sera traduit en une séquence d'acide aminé pour obtenir une protéine (Ameziane *et al.,* 2006).

6. Processus de la traduction

6.1. La traduction chez les procaryotes

6.1.1. Initiation

Chez les procaryotes l'initiation nécessite des facteurs spécifiques appelés facteur IF pour *Initiation Factors* (IF1.IF2.IF3) et chacun de ces facteurs a une fonction particulière.

La première étape de l'initiation consiste à trouver le codon start c'est-à-dire le premier triplet de nucléotides qui sera traduit. Ce dernier correspond en général à un codon AUG.

Pour reconnaitre le codon de démarrage AUG, l'extrémité 3' de l'ARN 16S de la petite sous-unité du ribosome, fait une interaction avec la séquence particulière située sur l'ARN messager juste en amant, cette dernière est appelée séquence SD (schine-dalgarno).

Le démarrage de la traduction se fait par l'association des ribosomes à un autre élément crucial. Il s'agit de l'ARNt. L'ARNt du premier codant est assez particulier. Il est appelé ARNt-FMet initiateur ; ce dernier porte la Formylméthionine.

L'ARNt-FMet va alors se fixer au niveau du site P de la petite sous unité 30 S du ribosome et il va se lier avec le codon AUG de l'ARNm.

Après l'assemblage et la stabilisation, la grande sous unité 50S va s'associer à la sous unité 30S pour former le complexe d'initiation de la traduction (Luchetta, 2009).

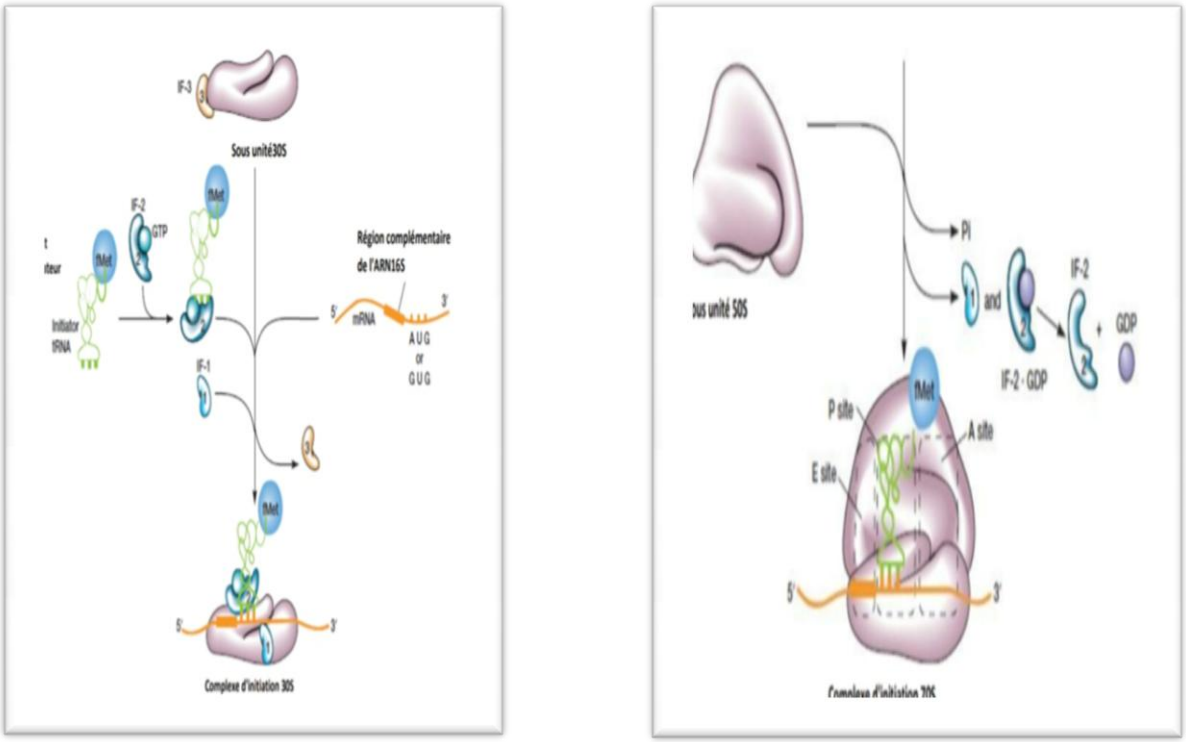


Figure 7. La phase d'initiation chez les procaryotes (Prescott, 2002).

6.1.2. L'élongation

Cette étape fait intervenir les facteurs d'élongation spécifiques appelés facteurs EF pour *Elongation Factors* (EF-tu, EF-ts, EF-G). Cette étape nécessite l'énergie sous forme de GTP.

Après la formation du complexe d'initiation le site E et libre, le site P est occupé par l'ARNt-fMet, le site A est libre (Luchetta, 2009).

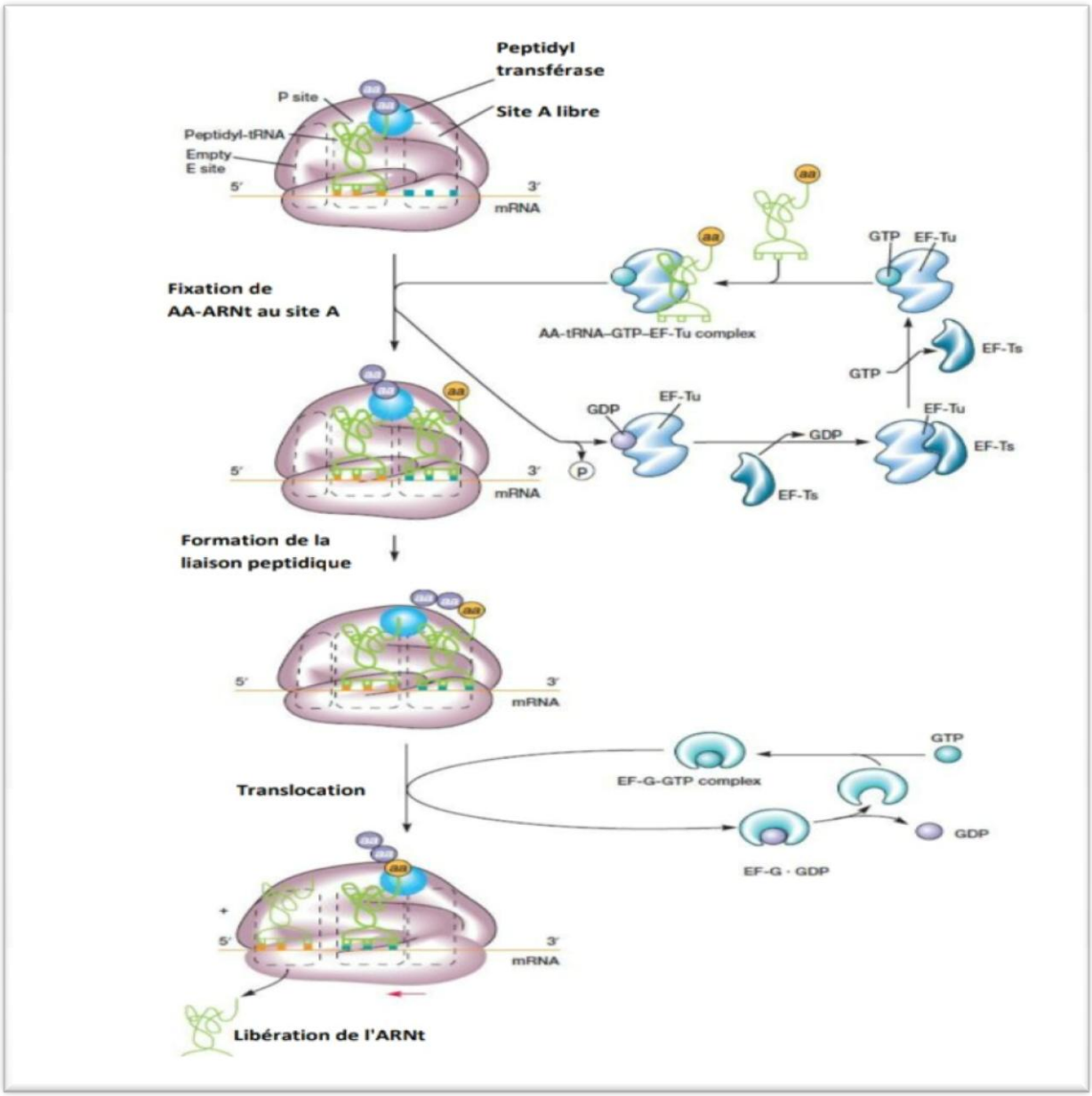


Figure 8. La phase d'élongation chez les procaryotes (Prescott, 2002).

6.1.3. Terminaison

Les facteurs de terminaison (RF) interviennent lorsque le site A du ribosome arrive à un codon stop. Ces facteurs sont en nombre de 4 : RF1, RF2, RF3 et RRF. Chacun d'eux possède une fonction particulière dans le processus de terminaison.

Les deux premiers peuvent entrer dans le site A à la place d'un ARNt car ils ont une structure 3D identique à celle d'un ARNt.

Les deux autres facteurs serviront à la dissociation de l'ensemble ribosome ARNt-ARNt-protéine (Luchetta, 2009).

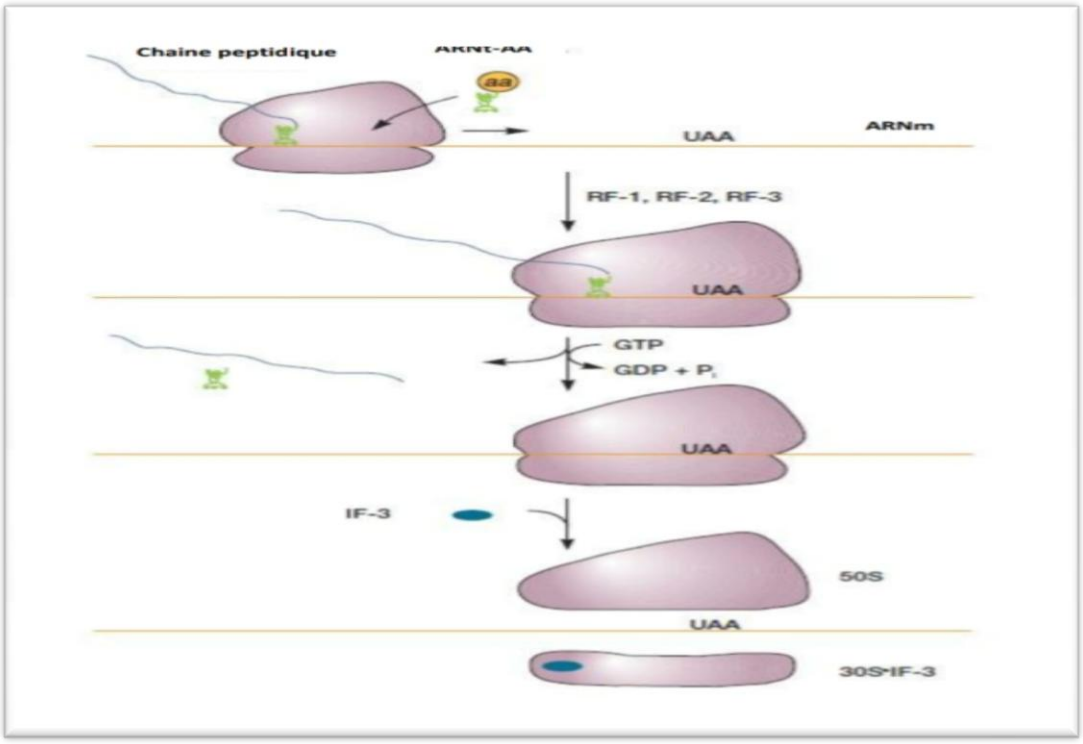


Figure 9. La phase de terminaison chez les procaryotes (Prescott, 2002).

6.2. La traduction chez les eucaryotes

6.2.1. Initiation

Contrairement au procaryotes l'initiation chez les eucaryotes est très complexe parce qu'elle nécessite 5 facteurs d'initiation eIF (de eIF-1 à eIF-5).

Le mécanisme de recherche du codon start est l'initiation par scanning à partir de l'extrémité 5' de l'ARNm.

Les ARNm possédant une coiffe en 5' (méthyl guanosine) et une séquence polyadénylée en 3'.

Le facteur de traduction eIF4 reconnait la coiffe et permet à la sous –unité 40S du ribosome de se fixer. Cette dernière se déplace sur l'ARNm pour la recherche du codon AUG puis le ribosome du site connait une stabilisation sur ce codon.

Le ribosome va se fixer au niveau du codon AUG et l'ARNt-FMet va se fixer au site P de la petite sous unité 40S.

Après la stabilisation la grande sous unité 60S se fixe et forme le complexe d'initiation (Luchetta, 2009).

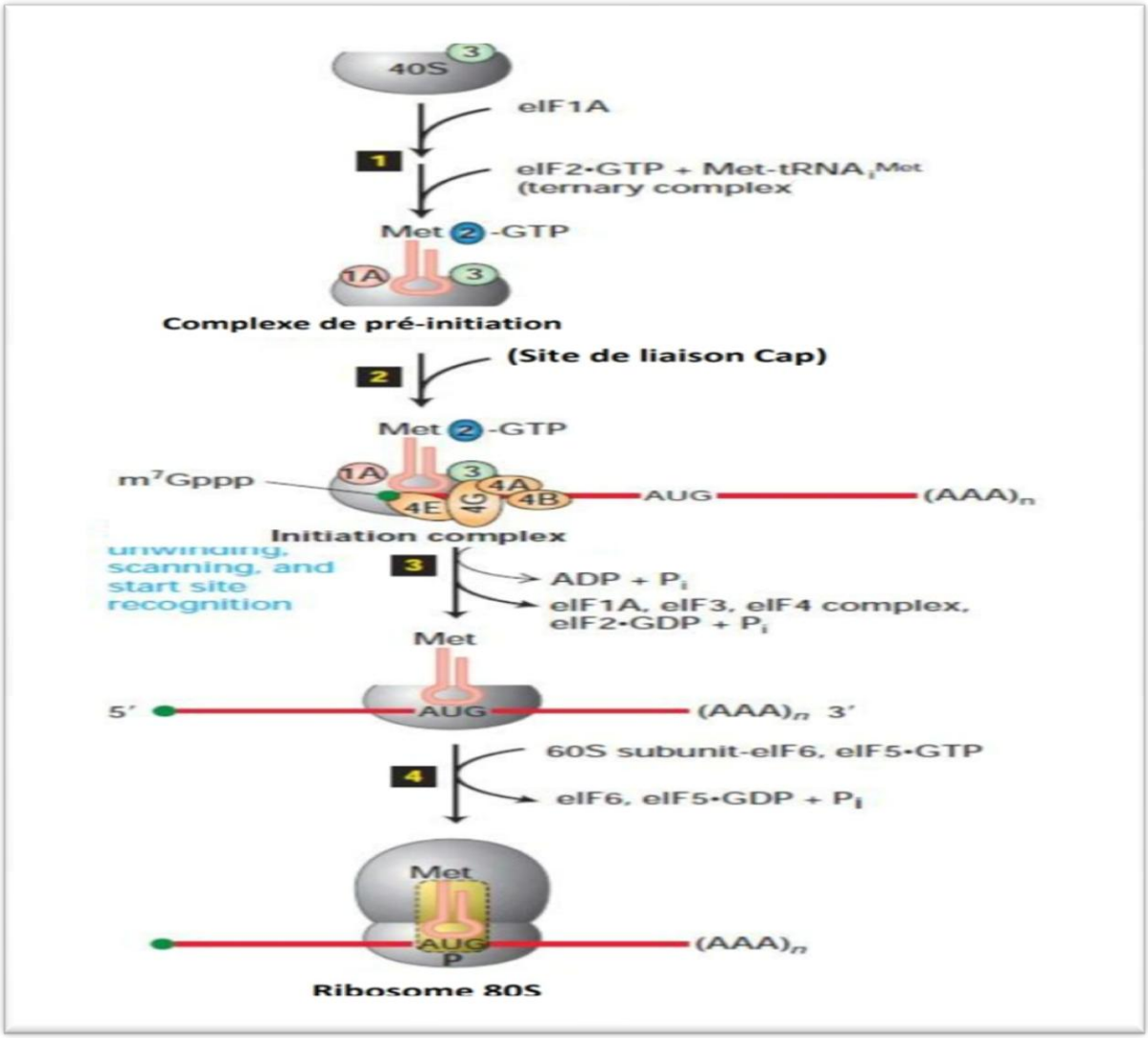


Figure 10. La phase d'initiation chez les eucaryotes (Lodish *et al.*, 2003).

6.2.2. L'élongation

Cette étape aura lieu après l'initiation de la traduction ; la présence des facteurs d'élongation est très importante, ces facteurs sont nommés les facteurs Eef (l'élongation factor).

L'élongation a besoin d'énergie sous forme de GTP. Ce mécanisme est très proche de celui observé chez les procaryotes (Luchetta, 2009).

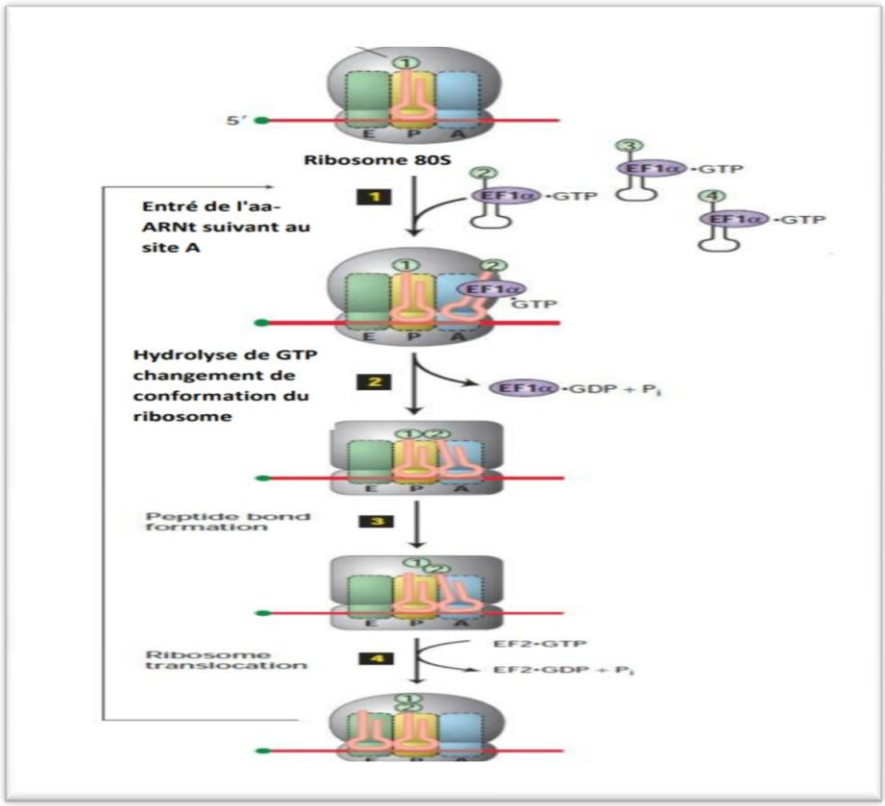


Figure 11. La phase d'élongation chez les eucaryotes (Lodish *et al.*, 2003).

6.2.3. Terminaison

On trouve ici aussi un mécanisme proche des procaryotes avec des facteurs de terminaison spécifiques pour les eucaryotes appelés eRF (eRF-1 et eRF-2) (Luchetta, 2009).

7. Séquence chimère

C'est une séquence d'ADN imaginaire qui n'existe pas dans la nature et dans les banques des données. Pour sa construction, cette séquence chimère a été construite selon les mêmes propriétés d'une séquence d'ADN qui existe dans la nature.

Chapitre : 02
Notions informatiques et bioinformatiques

1. Historique

L'informatique a été créée en 1962 par Philipe Dreyfus.Elle est composée de deux termes qui sont « information » et « automatique »(Breton, 1990).

La progression de l'informatique est associée au développement de la recherche en mathématiques et en particulier les algorithmes qui sont apparu au 9ème siècle avec les travaux du mathématicien arabe *Abu Jaffar Al Khawarizmi*. Le développement de l'informatique est lié aussi à l'introduction du calcul binaire en 1967 grâce aux travaux de*GottredWihem Leibniz*(Delmas, 2009).

Il est important de noter que l'informatique désigne l'ensemble des sciences et des techniques qui permettent le traitement automatique de l'information incluant aussi bienla partie matériel (électronique) et la partie immatérielle(logiciel) d'un ordinateur(Le Parc, 2017).

L'information est un renseignement qui possède un sens. Exemple : « la note est 10 » représente une information. Le renseignement c'est le numéro 10. Ce numéro détient le sens suivant : « le numéro 10 représente une valeur d'une note ».

Le traitement automatique est un ensemble des opérations réalisées par des moyens automatiques, relatif à la collecte, l'enregistrement, l'élaboration, la modification, la conservation, la destruction, l'édition de données et, d'une façon générale, leur exploitation (Makowski, 1981).

2. Notions informatiques

Dans ce que suit, nous présentons les notions informatiques qui peuvent être employées dans ce travail.

2.1. Algorithme

Un algorithme est une méthode qui permet de calculer et de résoudre une série de problèmes.Cette méthode nous permet d'exprimer la structure logiqued'un programme(Tisseau, 2009).

2.2. Structure des données

La meilleure façon d'organiser un ensemble de données dans le but d'y accéder rapidement (Académie française, 1966).

2.3. Programme

Un programme informatique est un enchainement des instructions écrites, selon un langage de programmation afin d'être exécutables par l'ordinateur et compréhensibles par l'humain (Jaunatre, 2014).

2.4. Modélisation

La modélisation est un processus par lequel on organise les connaissances portant sur un système donné (Zeigler*et al.*,2000).

La modélisation est une abstraction de la réalité, une description d'un système dynamique (Fishwick et Zeigler, 1976).

2.4.1. Définition d'un modèle

Le modèle est une simple représentation de la réalité pour réaliser un traitement dans un ordinateur et pour résoudre un problème d'analyse ou de conception (Duboz *et al.*,2004).

Le modèle permet donc de spécifier le système à réaliser, de valider le modèle vis-à-vis des clients, de fournir un guide pour la construction du système pour organiser les structures de données et le comportement du système, et de documenter le système et les décisions prises (Taconet*et al.*, 2015).

2.4.2. Les caractéristiques d'un modèle (Popper, 1969).

Le modèle est caractérisé par :

- Le modèle doit être ressemblé avec le système réel.
- Le modèle doit constituer une simplification du système réel.
- Le modèle doit être relié au monde réel.
- Un modèle peut être exprimé avec différents niveaux d'abstraction / raffinement
- Le modèle n'est pas la réalité, mais se construit à partir de l'observation de la réalité.

2.4.3. Le principe de modélisation

Modéliser c'est-à-dire abstraire la réalité pour mieux comprendre le système à réaliser.

Le processus de modélisation vise à obtenir une solution acceptable du système informatique. La solution finalement retenue n'est pas obtenue en une seule itération, maisplusieurs étapes sont nécessaires. Ces étapes successives permettent de raffiner le niveau de détails du système à réaliser.

Le processus de modélisation est nécessaire pendant toute la durée de vie du projet : discussion avec les clients ; analyse du système à réaliser, documentation commune entre les développeurs, etc.

La pérennité de l'information réalisée est un autre élément justifiant la décision de modéliser le système. En effet, le développement, mais aussi la maintenance corrective et la maintenance évolutive du système bénéficient de l'existence du modèle en tant que documentation de référence (Taconet*et al.*,2015).

2.5.Logiciel

On définit le logiciel comme un ensemble des programmes qui permettent à un système informatique d'assurer une tâche ou une fonction en particulier.Un logiciel représente un ensemble d'entité nécessaire au fonctionnement d'un processus de traitement automatique de l'information(Longuet, 2018).

2.5.1. Modèles de développement

Depuis le début des années 70, le domaine du génie logiciel développe des modèles visant à spécifier les différentes étapes de conception informatique d'un logiciel.

L'établissement de tels modèles a pour but de décomposer le processus de conception, de prévenir les éventuels défauts de conception, le plus en amont possible, et de faciliter la maintenance du système conçu.

Issus des premiers travaux dans ce domaine, des modèles ont été proposés pour la conception de touts types d'application. Le premier fut le modèle en cascade. L'étude de l'utilisation de ce modèle a mis en évidence la nécessité de prendre en compte la vérification et la validation des étapes dans les modèles de conception afin d'être le plus à même de détecter les erreurs le plus rapidement possible. Cette observation a donné naissance à la définition de plusieurs modèles raffinant le modèle en cascade comme le modèle en V. Ces modèles n'ont pas été définis pour exprimer la conception d'application interactive (Caffiau, 2006).

2.5.2.Les activitésdu cycle de vie d'un logiciel (Mondjo, 1961).

Le cycle de vie d'un logiciel est un ensemble des activités à suivre pour développer un logiciel. La manière d'appliquer ces activités suit un des modèles existants (en cascade, en v,...). Ces activités sont :

- **Spécification :**Décrit ce que doit faire le logiciel.
- **Conception** : Cette étape permet d'élaborer la structure générale du système et de définir chaque sous ensembles du logiciel à produire.
- **Implémentation** : C'est la réalisation du système. C'est programmer les fonctionnalités définies dans la phase de la conception en utilisant un langage de programmation.
- **vérification :**C'est une procédure permettant de vérifier le bon fonctionnement de chaque sous-ensemble du logiciel.
- **Validation :** Cette étape consiste à recueillir et à formaliser les besoins du client, de définir les contraintes et d'estimer la faisabilité de ces besoins.
- **Maintenance** : Cette étape permet de prendre en charge les actions collectives du système (maintenance et évolutive).

2.6.Les bases de données

Une base des données est tout simplement une collection d'informations diverses c'est-à-dire une collection des données structurées. Ces données sont enregistrées sur des supports accessibles par l'ordinateur.Les données structuréessont organisées de façon à retrouver facilement l'information souhaitée. La base des données représente des informations sur le monde réel.

Les bases de données ne sont pas identiques, elles diffèrent selon leurs utilisations ou selon les informations qui y sont enregistrées (Beroud, 2010).

2.6.1. Définition d'une banque de données

Une banque de données regroupe souvent plusieurs bases de données ; fréquemment les données sont stockées sous la forme de fichiers (Legrand, 2016).

2.6.2.La différence entre base et banques de donnée

Souvent les termes de banque ou base sont utilisés sans distinction particulière. Toutefois, il existe une différence non seulement pour l'utilisateur, mais aussi pour l'implémentation informatique de ces dernières **(**Mesguich et Normier, 1982).

❖ **Banque de données**

- Ensemble de données relatif à un domainedes connaissances défini.
- Structuration pas obligatoire.
- Volume important.

❖ **Base de données**

- Organisée en vue de son utilisation par des programmes.
- Structuration et modèles obligatoires.
- Volume indifférent.

3. Les bases de données biologiques

Les bases de données biologiques contiennent des informations biologiques. Ces dernières concernent des données homogènes spécifiques sur : une maladie, une espèce vivante, une molécule. Les bases de données biologiques sont donc spécifiques(Mahec, 2008).

3.1.Les types des bases biologiques

On distingue deux types qui sont :

3.1.1 Les bases généralistes

Ce sont des bases publiques internationales, qui stockent des séquences d'ADN ou de protéines déterminées quel que soit l'organisme ou la méthode utilisée pour les acquérir(Mahec, 2008). Ces bases sont reconnues par les banques de données biologiques.

3.1.2. Les bases spécialisées

Ce sont les bases qui peuvent classer les informations selon une caractéristique biologique particulière comme par exemple les bases recensant toutes les séquences d'un même génome (Mahec, 2008).

4. Les banques de données biologiques

Les banques de données biologiques contiennent des informations du domaine de la biologie. Ces données biologiques sont des séquences primaires d'ADN, ARN et de protéines.

Les banques de données biologiques sont consultables sur le Web.

L'obtention des séquences trouvées dans les banques se fait par plusieurs manières.Elles peuvent être isolées à partir d'une cellule, déduit à partir de la séquence nucléique par simple traduction ou encore par génie génétique(Meshoul, 2007).

4.1 Les types des banques de données biologiques (Meshoul, 2007).

4.1.1 Les banques des séquences nucléiques

Les banques les plus populaires sont :

- **EMBL :**banque européenne crée en 1980 et financée par l'EMBO (EuropeanMolecularyBiologyOrganization) (Hamm et Cameron, 1986).Elle est aujourd'hui diffusée par l'EBI (EuropeanBioinformatics Institute, Cambridge, UK).
- **GenBank :** créée en 1982 par la société IintelliGenitics et diffusée maintenant par le NCBI (National Center for Biotechnology Information, Los Alamos, US). (Bilofsky et Burks ,1988) Elle est soutenue par le NIH (National Institue of Health).Elle possède plus de 50 millions séquences stockées.
- **DDBJ :**(DNA Data Bank) : créée en 1986 et diffusée par le NIG (National Institute of Genetics, Japon).

4.1.2 Les banques de séquences protéiques

- **PIR-NBRF :** créée en 1984 par la NBRF (National BiomedicalResearchFoundation) Elle est maintenant un ensemble de données issues du MIPS (Martinsried Institute for Protein Séquences, Munich, Allemange) et de la banque japonaise JIPID (Japan International Protein Information Database)
- **SwissProt :** crée en 1886 à l'université de Genève et maintenue depuis 1987 dans le cadre d'une collaboration entre cette université (via ExPaSy, Expert ProteinAnalysis System) et l'EMBL .Celle-ci regroupe aussi des séquences annotées de la banque PIRNBRF ainsi que des séquences codantes, traduites de l'EMBL.

4.1.3. Les banques de motif

Sontdédiées aux stockages des motifs protéiques ayant une signification biologique (Hofmann *et al.,* 1999). Cette banque peut être considérée comme un dictionnaire de

motifs.Les bases de ce type ont pour mission le recensement dans des catalogues, les séquences des différents motifs pour lesquels une activité biologique a été identifiée.

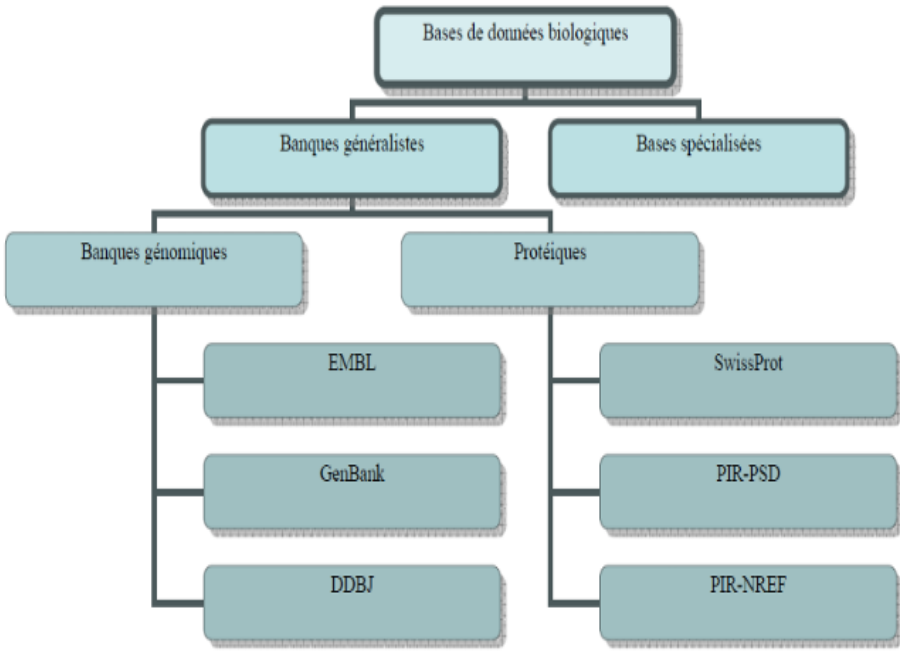


Figure 12. Illustration qui représente les types des bases biologiques.

4.2. L'importance des banques et bases des donnés biologiques (Mezhoud, 2004).

- Identification des protéines homologue.
- Horthologue : on cherche des organismes différents.
- Paralogue : organisme identique.
- Déterminer si des séquences ont une fonction similaire ou proche.
- Déterminer des familles des protéines ayant un domaine conservé.
- Localiser des régions codantes et non codantes.
- Etablir des relations entre les séquences.

5.Alignement des séquences

En bioinformatiques, l'opération d'alignement vise à identifier des zones communes à un groupe de K séquence.

Les zones qui se ressemblent sont dites similaires ou homologues si elles dérivent d'un ancêtre commun (Richer, 2008).

5.1.But de l'alignement

L'alignement permet de (Richer, 2008) :

- Identifier des motifs fonctionnels ou structurels conservés.
- Trouver deszones non conservées qui résultent d'événements spécifiques.
- Déterminer si des séquences ont divergé depuis un ancêtre commun.

5.1.Les types d'alignement (Richer, 2008).

5.1.1. Global

C'est un alignement entre deux séquences sur la totalité de leur longueur, cet alignement permet de mesurer le degré de similitude entre deux séquences**.**

5.1.2. Local

C'est un alignement qui permet d'identifier des similarités entre une séquence et une sous-séquence, c'est-à-dire des régions isolées**.**

Cet alignement nous permet de trouver des segments qui ont un haut degré de similitude**.**

5.1.3. Multiple

C'est un alignement global entre plusieurs séquences (plus de deux séquences)**.**

5.2. outils d'alignement des séquences

Nous présentons les outils que nous avons utilisés dans ce travail.

5.2.1. Basic Local Alignment Search Tool (BLAST)

BLAST est certainement l'application de bioinformatique la plus utilisée par les biologistesautour du monde. Cette application est utilisée pour la recherche de similitudes entre des séquences de nucléotidesou d'acides aminés.BLAST a pour rôle de tenter de déterminer la fonction d'un gène ou d'une protéine en comparaison avec des séquences dont le rôle est bien connu. Comme son nom l'indique, BLAST détecte les similarités locales entre deux séquences, similarités qui ont en général plus de sens d'un point de vue biologique que des similarités globales qui peuvent apparaitre entre des séquences aux rôles fort différents (Mahec, 2008).

D'autre part BLAST fait la traduction de façon indirecte, mais comment ?

Après avoir introduit la séquence nucléique dans le BLAST, ce dernier va chercher dans les banques de données une séquence qui ressemble a la séquence qui à été introduite. Puis, il va chercher dans les banques de données protéiques et il nous donne la séquence

peptidique adéquate. Donc, il fait la traduction de façon indirecte mais seulement pour les séquences naturelles et pas pour les séquences chimères.

5.2.2. CLUSTAL OMEGA

CLUSTAL OMEGA est un logiciel largement utilisé pour exécuter un alignement multiple. Il est développé il ya près de dix ans pour répondre au nombre croissant de séquences protéiques ou nucléiques et la nécessité de réaliser de grands alignements rapidement et avec précision. CLUSTAL OMEGA est composé de plusieurs types qui sont CLUSTALW et CLUSTAL X et qui sont les packages les plus largement utilisés pour la fabrication de plusieurs alignement des séquences.

CLUSTAL OMEGA présentait une méthode simple et rapide pour créer des arbres de guidages.Il s'agit de regrouper les séquences utilisées pour décider de l'ordre d'alignement au cours de la dernière phase d'alignement progressif. L'idée générale est de commencer par des alignements de deux séquences seulement, généralement les plus proches de l'ensemble de données.Ensuite, l'alignement construit à aligner les alignements les uns avec les autres ou on les alignant l'un après l'autre, en fonction de la topologie de l'arbre de guidage (Thompson *et al*., 1994).

6. Logiciels de traduction automatique des séquences ADN en séquences protéiques

Il existe plusieurs logiciels qui font la modélisation et la simulation du processus de la synthèse des protéines à partir d'une séquence d'ADN d'un gène. Parmi eux, le logiciel de la traduction automatique Anagène.

Grace à ses contacts réguliers avec des laboratoires à Orsay et Villejuif, l'équipe de l'INRP(Institut National de la Recherche Pédagogique)s'est vu conseiller le logiciel SQAID par un chercheur d'Orsay (jean-Lous Risler).SEQAID est un logiciel américain gratuit tournant sous DOS pour analyser des séquences nucléiques et protéiques. Par la suite, l'équipe de l'INRP est entrée en contact avec les auteurs de SEQAID (Rhoads et Roufa) pour avoir l'autorisation de traduire le logiciel en français.

En 1997, Anagène a remplacé SEQAID. Eneffet, Windows prenant de l'ampleur, il était intéressant d'avoir un logiciel qui ne tourne pas sous DOS.

Anagène a été construit par l'INRP et le CNDP strictement à partir des fonctions générales de SEQAID et enrichi avec des fonctions scientifiques.

En 2006, ladernière version d'Anagène est sortie et intégrait de nouvelles séquences (Stanislas, 2011).

Chapitre : 03

Matériels et méthodes

1. Introduction

Comme nous avons dit dans le premier chapitre, la production des protéines à partir d'une séquence d'ADN passe par plusieurs étapes.

La première étape est la transcription. Cette dernière consiste à transférer l'information génétique sur une molécule d'ARNm, qui va subir quelques modifications afin de devenir un ARNm mature.

Enfin l'information génétique contenue dans l'ARNm mature va être traduite en protéine.

Alors, comment peut-on faire la modélisation de ce processus naturelle par l'utilisation d'un modèle informatique ? En d'autres termes, comment développer un logiciel qui permet de « mimer » ce processus.

Pour cela, nous avons choisi d'utiliser un modèle qui est le modèle en cascade. Selon ce modèle, Le cycle de vie d'un logiciel représente toutes les étapes de son développement d'une manière séquentielle depuis la spécification et l'analyse des données, la conception et l'implémentation puis la validation, jusqu'à la vérification.

Nous avons exécuté le logiciel développé, après l'implémentation dans le langage MATLAB, sur des séquences naturelles et qui existent dans les banques des données. Puis, nous avons proposé aussi une séquence « chimère » imaginaire, et nous avons confirmé qu'elle n'a pas de semblable dans les banques de données (GenBank et EMBL). Ensuite, nous avons exécuté le logiciel sur cette séquence chimère.

2. Spécification

Cette étape nous permet d'élaborer et d'expliquer informellement l'architecture de ce logiciel. On va expliquer l'enchaînement des différentes phases du processus naturel, le fonctionnement de chaque phase, les entrées et les sorties de chaque phase.

Comme le processus naturel, le logiciel va être composé de trois fonctions. Chaque fonction permet de modéliser une phase du processus naturel. Les phases sont dans l'ordre suivant :

2.1. La transcription

Cette fonction permet de modéliser la phase de la transcription dans le processus naturel. Elle possède comme entrée la séquence ADN qui représente le brin complémentaire de la partie codante du gène et elle va donner comme sortie la séquence ARN messagé. La

partie codante est une séquence ADN qui va être traduite en séquence d'acides aminés (protéine) et qui vérifie les conditions suivantes :

- La taille de cette séquence (le nombre des bases) doit être supérieure ou égal à 300 pb
- Les trois premières bases de cette séquence sont AUG qui correspond au codon start. Ce codon doit code l'acide aminé Méthionine (MET) et doit obligatoirement être présent.
- Le nombre des bases de cette séquence est un nombre diviseur du nombre 3
- Les trois dernières bases de cette séquence sont soit TAA ou TAG ou TGA qui correspondent au codon stop.

Le gène est une partie de la séquence ADN. La séquence ADN est composée de deux brins : brin principal et brin complémentaire. Le brin complémentaire est composé des bases complémentaires des bases qui composent le brin principal de telle sorte :

- Base A(Adénine) est le complément de la base T (Thymine)
- Base T (Thymine) est le complément de la base A(Adénine)
- Base G(Guanine) est le complément de la base C (Cytosine)
- Base C(Cytosine) est le complément de la base G (Guanine)

Une protéine est constituée de plus de 100 acides aminés. Une chaîne de moins de 100 acides aminés est appelée Polypeptide.

Le principe de fonctionnement de cette phase permet de construire la séquence ARN messagé à partir de la séquence ADN. Pour ce faire, toutes les bases T (Thymine) vont être transformées en base U (Uracile). Les autres bases restent sans changement.

2.2. La maturation

Cette fonction permet de modéliser la phase de la maturation dans le processus naturel. Cette phase possède comme entrée la séquence ARN (obtenue par la première fonction) elle va donner comme sortie la séquence ARNm mature. Le principe de fonctionnement de cette phase est de construire ARNm mature à partir de l'ARN par la décomposition de l'ARN en un ensemble des codons où chaque codon est composé de trois bases.

2.3. La traduction

Cette fonction permet de modéliser la phase de la traduction dans le processus naturel. Cette phase possède comme entrée la séquence d'ARN mature (obtenue par la deuxième fonction) elle va donner comme sortie la séquence protéine (séquence des acides aminés). Le principe de fonctionnement de cette phase est de construire la séquence protéique à partir de l'ARN mature par la traduction de chaque codon en un acide aminé selon le tableau suivant :

Tableau 1. Les codons ADN des acides aminés et leurs abréviations (Mehdi et Meziani, 2018).

Amino Acid	Abbreviation 3-Lettres	Abbreviation 1 -Lettre	Codon(s)
Alanine	Ala	A	GCA, GCC, GCG, GCT
Arginine	Arg	R	CGA, CGC, CGG, CGT, AGA, AGG
Aspartic acid	Asp	D	GAC, GAT
Asparagine	Asn	N	AAC, AAT
Cysteine	Cys	C	TGC, TGT
Glutamic acid	Glu	E	GAA, GAG
Glutamine	Gln	Q	CAA, CAG
Glycine	Gly	G	GGA, GGC, GGG, GGT
Histidine	His	H	CAC, CAT
Isoleucine	Ile	I	ATA, ATC, ATT
Leucine	Leu	L	CTA, CTC, CTG, CTT, TTA, TTG
Lysine	Lys	K	AAA, AAG
Methionine	Met	M	ATG
Phenylalanine	Phe	F	TTC, TTT
Proline	Pro	P	CCA, CCC, CCG, CCT
Serine	Ser	S	TCA, TCC, TCG, TCT, AGC, AGT
Threonine	Thr	T	ACT, ACC, ACG, ACT
Tryptophan	Trp	W	TGG
Tyrosine	Tyr	Y	TAC, TAT
Valine	Val	V	GTA, GTC, GTG, GTT
STOP	-	-	TAG, TAA, TGA

3. Conception

Cette phase consiste à élaborer l'explication formelle de l'architecture de ce logiciel.

1- Chaque base azotée (A, G, T, C, U) est modélisée en informatique par un caractère.
2- Chaque acide aminé peut être modélisé par un caractère si on exploite l'abréviation en une seule lettre ou par une chaine de caractères si on exploite l'abréviation en trois lettres.
3- Les séquences ADN et ARN vont être modélisées par des chaines de caractères.
4- La séquence d'ARN mature est modélisée par une matrice de caractères, car l'ARN mature est composée d'un ensemble de codons où chaque codon est modélisé par une chaine de 3 caractères.
5- La Protéine est modélisée par un tableau de caractères dans le cas où on exploite l'abréviation en une seule lettre pour chaque acide aminé.
6- La Protéine est modélisée par une matrice de caractères dans le cas où on exploite l'abréviation en trois lettres pour chaque acide aminé.

Dans ce qui suit, nous expliquons l'algorithme de chaque fonction.

3.1. La transcription

Si la taille de la chaine de caractère d'ADN modulo 3 différent de 0
Ecrire « la traduction est impossible »
Si non
Parcourir la chaine ADN caractère par caractère jusqu'à la fin
Si le caractère est « t » ou « T » alors
Ecrire dans ARN « U »
Si le caractère est « a » ou « A » alors
Ecrire dans ARN « A »
Si le caractère est « c » ou « C » alors
Ecrire dans ARN « C »
Si le caractère est « g » ou « G » alors
Ecrire dans ARN « G »

3.2. La maturation

Soit la variable i qui modélise un compteur qui compte le nombre des éléments dans la chaine d'ARN.

Soit la variable j qui modélise un compteur qui compte le nombre des lignes dans la matrice d'ARN mature.

Parcourir la chaine d'ARN jusqu'à la fin (prendre 3 caractères pour chaque itération)

Mettre les trois caractères de l'ARN dans la même ligne de l'ARN mature, chaque caractère dans une colonne.

Incrémenter i par 3 (i=i+3)

Incrémenter j par 1 (j=j+1)

3.3. La traduction

Un dictionnaire, qui permet d'affecter à chaque acide aminé le codon correspondant, est modélisé par une matrice de chaine de caractères. Dans chaque ligne, les premières cases comportent le code de l'acide aminé et les trois dernières cases comportent les trois bases constituant le codon.

Lire la première ligne de la matrice d'ARN mature pour obtenir le premier codon

Chercher dans la matrice dictionnaire si ce codon correspond à l'acide aminé MET (Méthionine)

S'il ne correspond pas

Ecrire « la traduction est impossible »

S'il correspond alors

Ecrire le code de l'acide aminé MET (Méthionine) dans la première ligne du tableau/matrice protéine.

Parcourir la chaine d'ARN mature, à partir de la deuxième ligne, ligne par ligne jusqu'à la fin pour obtenir chaque codon ou jusqu'à trouver un codon stop

Chercher dans la matrice dictionnaire l'acide aminé qui correspond à ce codon

Ecrire le code de l'acide aminé obtenu dans la ligne correspondante du tableau/matrice protéine.

Incrémenter par 1 le compteur qui compte les lignes de la protéine.

Nous avons fait une modélisation de cette séquence, on a proposé des séquences d'ADN sous forme d'un tableau de caractères qui va transformer les séquences d'ADN en ARN, c'est la phase de transcription dans le processus naturelle. Cette dernière doit subir une

maturation c'est pour cela on a fait une matrice des séquences ARN en ensemble des triplets donc l'ARNm devient mature ; c'est la phase de maturation dans un processus naturel.

La séquence d'ARNm mature est traduite en protéine, on a formulé alors une matrice qui contient les noms des acides aminés et une autre contient leur code génétique.

Finalement on a vérifié le codon stop qui indique la fin de la traduction.

D'autre part la séquence chimère que nous avons proposée doit être :

- d'une taille supérieure à 300 pb
- Possédant un nombre de base diviseur du nombre 3
- Commençant obligatoirement par une MET.
- Commençant par un codant start est se terminant par un codon stop.

Cette séquence suit les mêmes étapes de modélisation par le logiciel afin de produire une protéine.

La dernière étape consiste à vérifier et comparer les séquences protéiques réelles avec les séquences protéiques obtenues après la modélisation pour maitriser la qualité du logiciel.

Concernant l'alignement on doit vérifier la similarité entre notre séquence chimère et toutes les séquences nucléiques qui sont trouvées dans le BLAST.

4. Implémentation

Afin que la modélisation sous forme d'un algorithme que nous avons développé précédemment, puisse être exécutable par l'ordinateur, il est nécessaire de la traduire dans un langage de programmation. Nous avons choisi le langage MATLAB, car c'est le seul langage que nous avons étudié durant notre parcours d'une part et il est le langage le plus adéquat pour l'exploitation des tableaux et des matrices d'autre part.

4.1. MATLAB

Le langage MATLAB est dérivé de l'anglais Matrix Laboratory. C'est un langage de calcul vectoriel et matriciel. Il permet de réaliser des simulations numériques basées sur des algorithmes d'analyse numérique, de manipuler des matrices, d'afficher des courbes et des données et permet aussi de résoudre des problèmes de calcul très complexes d'une façon simple et rapide (Marie Postel, 2004).

L'objectif de ce langage est de développer des prototypes des logiciels et de tester de nouveaux algorithmes.

4.2. L'implémentation des fonctions du logiciel développé en MATLAB

Dans ce qui suit, nous présentons l'implémentation des fonctions vues dans la section précédente.

a. Fonction transcription

La figure suivante représente l'implémentation de cette fonction en MATLAB.

```
function [ARN] = transcription1 (ADN)
    y=size(ADN);
    if y(1)==1
        y1=y(2);
    else
        y1=y(1);
    end
    for i=1:y1

        if (ADN(i)=='T' || ADN(i)=='t')
           ARN(i)='U';
        else
          ARN(i)=ADN(i);
        end
```

Figure 13. Une représentation d'un extrait de l'implémentation de la fonction transcription d'ADN en MATLAB

b. Fonction de maturation

La figure suivante représente l'implémentation de cette fonction en MATLAB.

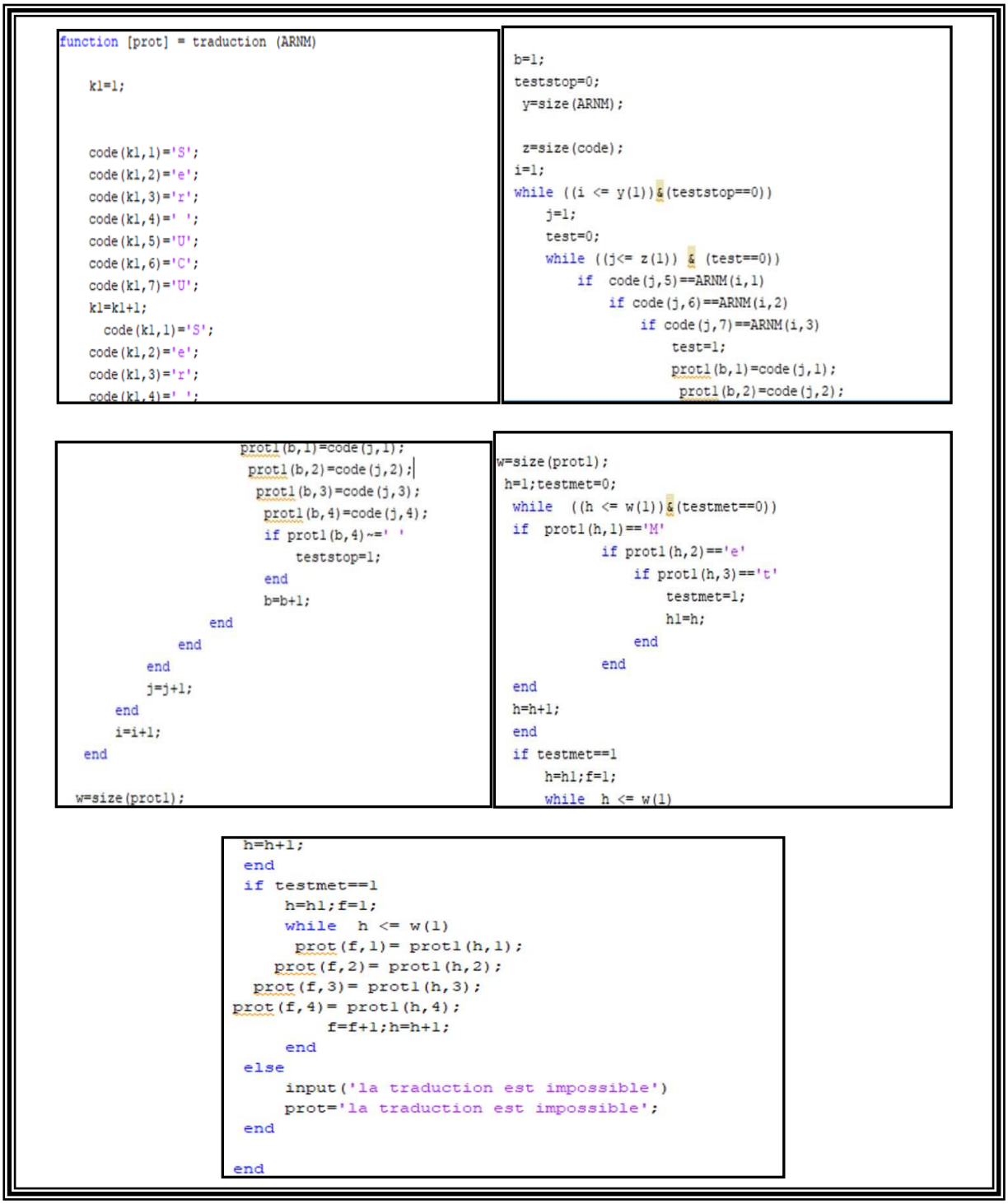
```
function [ARNM] = maturation (ARN)
    y=size(ARN);
    if y(1)==1
        y1=y(2);
    else
        y1=y(1);
    end
    y2=y1/3;
    j=1;
    i=1;
    u=y1;
    while i<=y1


     ARNM(j,1)=ARN(i);
      ARNM(j,2)=ARN(i+1);
       ARNM(j,3)=ARN(i+2);
```

Figure 14. Une représentation d'un extrait de l'implémentation de la fonction maturation d'ADN en MATLAB.

c. Fonction traduction

La figure suivante représente l'implémentation de cette fonction en MATLAB.



```
function [prot] = traduction (ARNM)

   k1=1;


   code(k1,1)='S';
   code(k1,2)='e';
   code(k1,3)='r';
   code(k1,4)=' ';
   code(k1,5)='U';
   code(k1,6)='C';
   code(k1,7)='U';
   k1=k1+1;
     code(k1,1)='S';
   code(k1,2)='e';
   code(k1,3)='r';
   code(k1,4)=' ';
```

```
b=1;
teststop=0;
 y=size(ARNM);

 z=size(code);
i=1;
while ((i <= y(1))&(teststop==0))
    j=1;
    test=0;
    while ((j<= z(1)) & (test==0))
        if  code(j,5)==ARNM(i,1)
            if code(j,6)==ARNM(i,2)
                if code(j,7)==ARNM(i,3)
                    test=1;
                    prot1(b,1)=code(j,1);
                     prot1(b,2)=code(j,2);
```

```
                    prot1(b,1)=code(j,1);
                     prot1(b,2)=code(j,2);
                      prot1(b,3)=code(j,3);
                       prot1(b,4)=code(j,4);
                       if prot1(b,4)~=' '
                           teststop=1;
                       end
                       b=b+1;
                end
            end
        end
        j=j+1;
    end
    i=i+1;
end

w=size(prot1);
```

```
w=size(prot1);
h=1;testmet=0;
 while  ((h <= w(1))&(testmet==0))
 if  prot1(h,1)=='M'
            if prot1(h,2)=='e'
                if prot1(h,3)=='t'
                    testmet=1;
                    h1=h;
                end
            end
 end
 h=h+1;
 end
 if testmet==1
     h=h1;f=1;
     while  h <= w(1)
```

```
 h=h+1;
 end
 if testmet==1
     h=h1;f=1;
     while  h <= w(1)
      prot(f,1)= prot1(h,1);
    prot(f,2)= prot1(h,2);
  prot(f,3)= prot1(h,3);
prot(f,4)= prot1(h,4);
          f=f+1;h=h+1;
     end
 else
     input('la traduction est impossible')
     prot='la traduction est impossible';
 end

end
```

Figure 15. Une représentation des extraits de l'implémentation de la fonction traduction d'ARNm mature en MATLAB.

d. Fonction codage

Cette fonction permet d'afficher les acides aminés selon l'abréviation en une seule lettre.

La figure suivante représente l'implémentation de cette fonction en MATLAB.

```
function [code] = codage(pro)
    y=size(pro);
    if y(1)==1
        y1=y(2);
    else
        y1=y(1);
    end
j=1;
    for i=1:y1

if (pro(i,1)=='S'& pro(i,2)=='e'& pro(i,3)=='r')
code(j)='S';
end

if (pro(i,1)=='P'& pro(i,2)=='h'& pro(i,3)=='e')
code(j)='F';
end
```

```
code(j)='F';
end


if (pro(i,1)=='L'& pro(i,2)=='e'& pro(i,3)=='u')
code(j)='L';
end


if (pro(i,1)=='I'& pro(i,2)=='l'& pro(i,3)=='e')
code(j)='I';
end

if (pro(i,1)=='M'& pro(i,2)=='e'& pro(i,3)=='t')
code(j)='M';
end
```

```
end

if (pro(i,1)=='V'& pro(i,2)=='a'& pro(i,3)=='l')
code(j)='V';
end

if (pro(i,1)=='P'& pro(i,2)=='r'& pro(i,3)=='o')
code(j)='P';
end

if (pro(i,1)=='T'& pro(i,2)=='h'& pro(i,3)=='r')
code(j)='T';
end

if (pro(i,1)=='A'& pro(i,2)=='l'& pro(i,3)=='a')
code(j)='A';
```

```
code(j)='A';
end


if (pro(i,1)=='T'& pro(i,2)=='y'& pro(i,3)=='r')
code(j)='Y';
end

if (pro(i,1)=='H'& pro(i,2)=='i'& pro(i,3)=='s')
code(j)='H';
end

if (pro(i,1)=='G'& pro(i,2)=='l'& pro(i,3)=='n')
code(j)='Q';
end
```

Figure 16. Une représentation des extraits de l'implémentation de la fonction codage en MATLAB.

e. Fonction globale

Après avoir terminé, nous avons créé une fonction globale qui permet d'appeler les autres fonctions séquentiellement. Elle prend comme entrée (input argument) la séquence ADN et elle donne comme résultat (output argument) la séquence protéique. La figure suivante représente l'implémentation de cette fonction en MATLAB.

```
function [ v4 ] = fonctionglobalm1( ADN )
%UNTITLED Summary of this lobalfunction goes here
%   Detailed explanation goes here
%v1=transcriptionl(ADN)
%v2=maturation(v1)
v1=transcriptionl (ADN);
    v2=maturation (v1);
    display('la séquence pro codée en 3e')
    v3=traduction (v2)
    display('la séquence pro codée en 1 symbole')
    v4=codage(v3)
end
```

Figure 17. Une représentation d'un extrait de l'implémentation de la fonction globale en MATLAB qui englobe les fonctions de (Transcription, maturation, traduction, codage).

5. Exécution

Nous avons exécuté le logiciel développé, après l'implémentation dans le langage MATLAB, sur des séquences naturelles et existantes dans les banques de données. Nous citons comme exemple la séquence « Alkalin Phosphatase ». Après avoir obtenu la séquence protéique en utilisant le logiciel développé, cette séquence va être alignée avec la séquence protéique issue de la banque EMBL, qui représente la traduction de « Alkalin Phosphatase ». Cet alignement donne un pourcentage d'identité de 100% qui prouve que le logiciel fonctionne correctement.

Puis, nous avons proposé aussi une séquence « chimère » imaginaire qui supposée ne pas exister dans les banques de données comme GenBank et EMBL. On utilise le logiciel BLAST pour prouver que cette séquence n'est pas identifiée (elle n'existe pas dans les banques et par conséquent elle n'est encore pas retrouvée dans la réalité).

Ensuite, nous avons exécuté le logiciel sur cette séquence chimère. Après avoir obtenu la séquence protéique en utilisant le logiciel développé, cette séquence va être blastée en utilisant le logiciel BLAST afin d'identifier cette séquence protéique (vérifier si elle existe dans la réalité).

Les résultats vont être discutés dans le chapitre suivant.

Chapitre : 04

Résultats et discussions

1. Validation et vérification des résultats

Après avoir appliqué certaines étapes de développement d'un logiciel, selon le modèle en cascade dans le chapitre précédent, nous continuons d'appliquer les étapes restantes dans ce chapitre. Il s'agit de valider et vérifier le logiciel produit.

La validation est une activité qui permet d'assurer que le logiciel fonctionne conformément à un ensemble d'exigences, de spécification et de besoins de l'utilisateur.

La vérification est une activité qui permet d'assurer que le logiciel ne donne pas de résultats erronés.

1.1. La vérification

La vérification est activité qui permet d'assurer que le logiciel ne donne pas de résultats erronés.

Donc, pour vérifier la qualité du logiciel, nous choisissons de l'exécuter sur une séquence nucléique naturelle (par exemple Alkalin phosphatase) qui est issue à partir d'une banque comme par exemple l'EMBL (European Molecular Biology Laboratory).

La figure suivante représente l'interface de la banque EMBL.

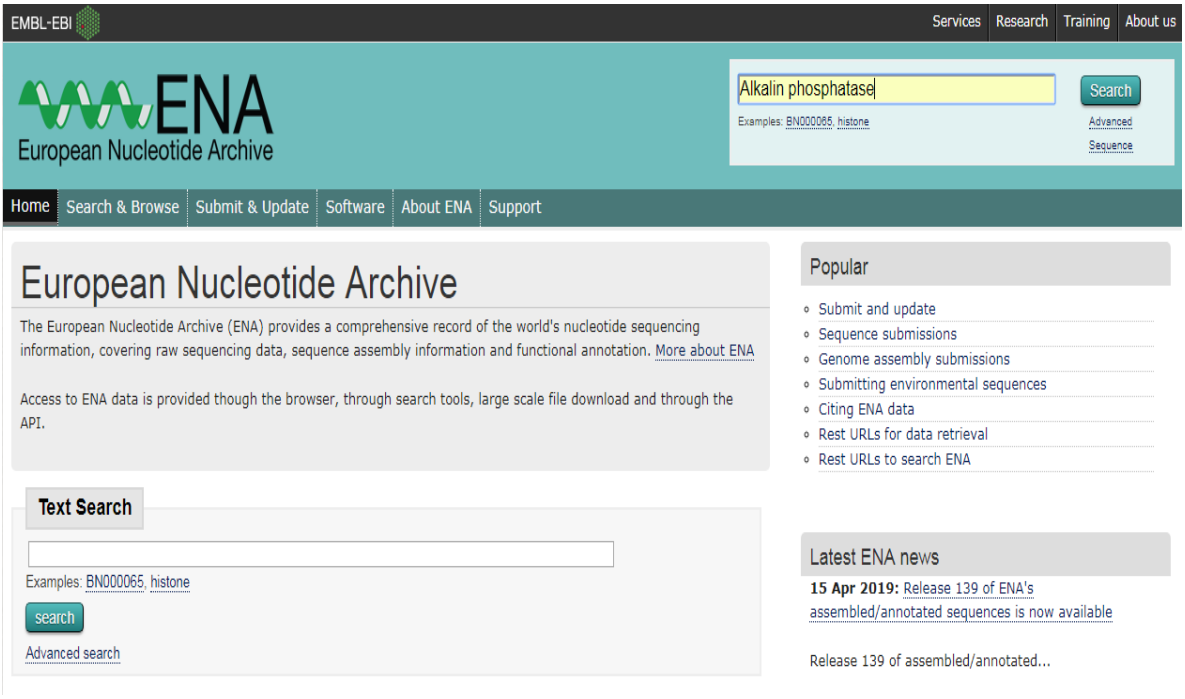


Figure 18. Fiche descriptive de la banque des données EMBL.

Nous prenons la partie codante de la séquence Alkalin phosphatase comme elle est indiquée dans la figure suivante.

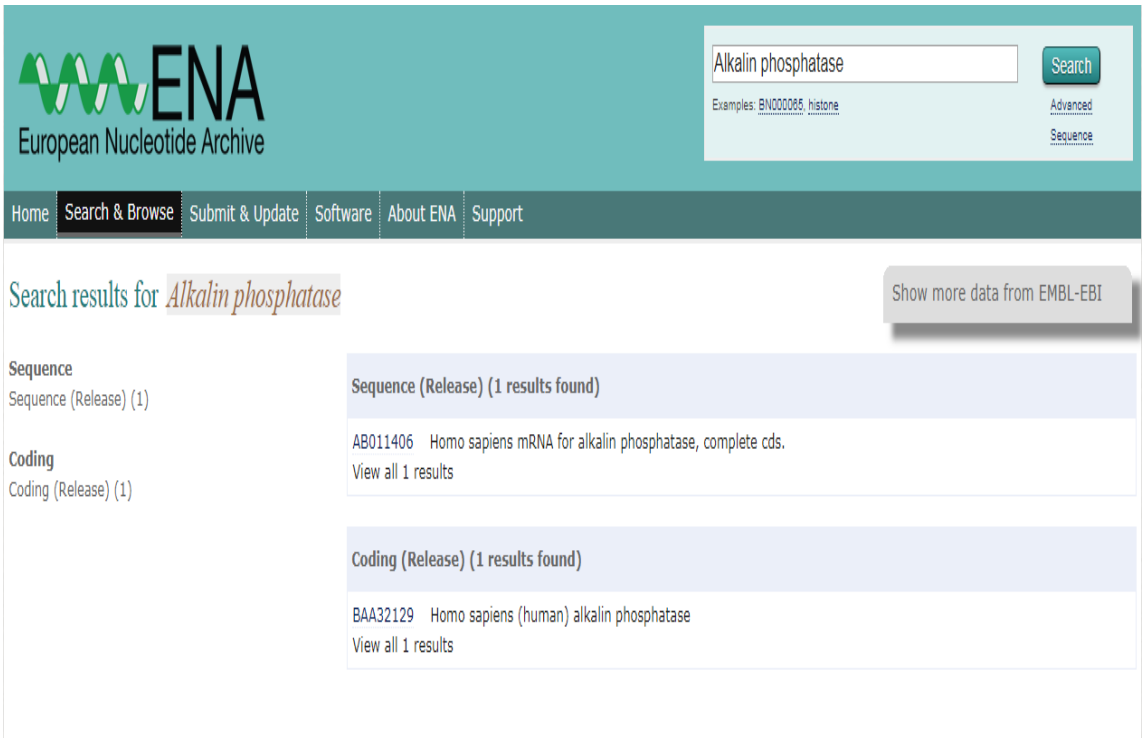


Figure 19. Fiche descriptive de la banque des données EMBL, qui indique la partie codante de la séquence d'Alkalin phosphatase.

Nous sommes assurés que cette partie codante commence par le codon ATG qui est le codon initiateur de la traduction. Ce dernier code pour la méthionine.

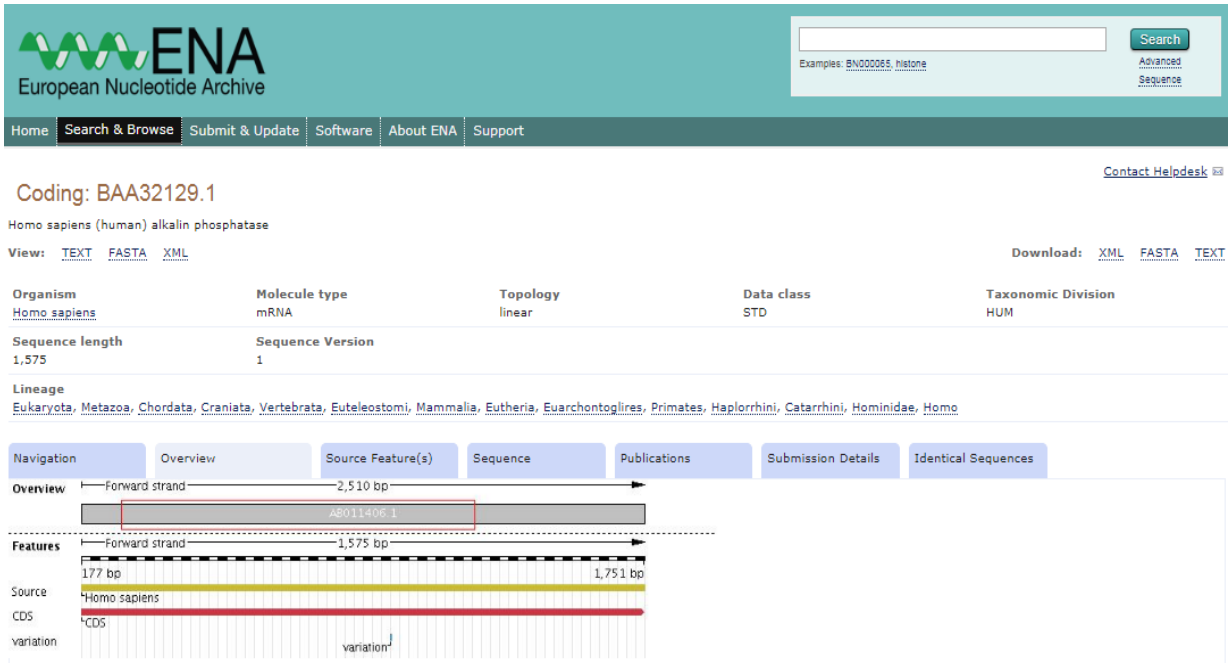


Figure 20. Illustration représente visualisation graphique de la partie codante.

La figure suivante présente la partie codante de la séquence Alkalin phosphatase écrite en forma Fasta.

Figure 21. La partie codante de la séquence Alkalin phosphatase écrite en forma Fasta.

Cette séquence a été utilisée comme une donnée d'entrée pour le logiciel développé sous forme d'une chaine de caractères comme suit :

ADN='ATGATTTCACCATTCTTAGTACTGGCCATTGGCACCTGCCTTACTAACTCCTTAGTGCCAGAGAAAGAGA
AAGACCCCAAGTACTGGCGAGACCAAGCGCAAGAGACACTGAAATATGCCCTGGAGCTTCAGAAGCTCAACA
CCAACGTGGCTAAGAATGTCATCATGTTCCTGGGAGATGGGATGGGTGTCTCCACAGTGACGGCTGCCCGCAT
CCTCAAGGGTCAGCTCCACCACAACCCTGGGGAGGAGACCAGGCTGGAGATGGACAAGTTCCCCTTCGTGGC
CCTCTCCAAGACGTACAACACCAATGCCCAGGTCCCTGACAGCGCCGGCACCGCCACCGCCTACCTGTGTGGG
GTGAAGGCCAATGAGGGCACCGTGGGGGTAAGCGCAGCCACTGAGCGTTCCCGGTGCAACACCACCCAGGG
GAACGAGGTCACCTCCATCCTGCGCTGGGCCAAGGACGCTGGGAAATCTGTGGGCATTGTGACCACCACGAG
AGTGAACCATGCCACCCCCAGCGCCGCCTACGCCCACTCGGCTGACCGGGACTGGTACTCAGACAACGAGATG
CCCCCTGAGGCCTTGAGCCAGGGCTGTAAGGACATCGCCTACCAGCTCATGCATAACATCAGGGACATTGACG
TGATCATGGGGGGTGGCCGGAAATACATGTACCCCAAGAATAAAACTGATGTGGAGTATGAGAGTGACGAGA
AAGCCAGGGGCACGAGGCTGGACGGCCTGGACCTCGTTGACACCTGGAAGAGCTTCAAACCGAGATACAAGC
ACTCCCACTTCATCTGGAACCGCACGGAACTCCTGACCCTTGACCCCCACAATGTGGACTACCTATTGGGTTTCT
TCGAGCCAGGGGACATGCAGTACGAGCTGAACAGGAACAACGTGACGGACCCGTCACTCTCCGAGATGGTGG
TGGTGGCCATCCAGATCCTGCGGAAGAACCCCAAAGGCTTCTTCTTGCTGGTGGAAGGAGGCAGAATTGACCA
CGGGCACCATGAAGGAAAAGCCAAGCAGGCCCTGCATGAGGCGGTGGAGATGGACCGGGCCATCGGCCACG
CAGGCAGCTTGACCTCCTCGGAAGACACTCTGACCGTGGTCACTGCGGACCATTCCCACGTCTTCACATTTGGT
GGATACACCCCCCGTGGCAACTCTATCTTTGGTCTGGCCCCCATGCTGAGTGACACAGACAAGAAGCCCTTCAC
TGCCATCCTGTATGGCAATGGGCCTGGCTACAAGGTGGTGGGCGGTGAACGAGAGAATGTCTCCATGGTGGA
CTATGCTCACAACAACTACCAGGCGCAGTCTCCTGTGCCCCTGCGCCACGAGACCCACGGCGGGGAGGACGTG
GCCGTCTTCTCCAAGGGCCCCATGGCGCACCTGCTGCACGGCGTCCACGAGCAGAACTACGTCCCCCACGTGA
TGGCGTATGCAGCCTGCATCGGGGCCAACCTCGGCCACTGTGCTCCTGCCAGCTCGGCAGGCAGCCTTGCTGC
AGGCCCCCTGCTGCTCGCTCTGGCCCTCTACCCCCTGAGCGTCCTGTTCTGA'

ADN =

'ATGATTTCACCATTCTTAGTACTGGCCATTGGCACCTGCCTTACTAACTCCTTAGTGCCAGAGAAAGAGAAAGA
CCCCAAGTACTGGCGAGACCAAGCGCAAGAGACACTGAAATATGCCCTGGAGCTTCAGAAGCTCAACACCAAC
GTGGCTAAGAATGTCATCATGTTCCTGGGAGATGGGATGGGTGTCTCCACAGTGACGGCTGCCCGCATCCTCA
AGGGTCAGCTCCACCACAACCCTGGGGAGGAGACCAGGCTGGAGATGGACAAGTTCCCCTTCGTGGCCCTCTC
CAAGACGTACAACACCAATGCCCAGGTCCCTGACAGCGCCGGCACCGCCACCGCCTACCTGTGTGGGGTGAAG
GCCAATGAGGGCACCGTGGGGGTAAGCGCAGCCACTGAGCGTTCCCGGTGCAACACCACCCAGGGGAACGA
GGTCACCTCCATCCTGCGCTGGGCCAAGGACGCTGGGAAATCTGTGGGCATTGTGACCACCACGAGAGTGAA
CCATGCCACCCCCAGCGCCGCCTACGCCCACTCGGCTGACCGGGACTGGTACTCAGACAACGAGATGCCCCCT
GAGGCCTTGAGCCAGGGCTGTAAGGACATCGCCTACCAGCTCATGCATAACATCAGGGACATTGACGTGATCA
TGGGGGGTGGCCGGAAATACATGTACCCCAAGAATAAAACTGATGTGGAGTATGAGAGTGACGAGAAAGCC
AGGGGCACGAGGCTGGACGGCCTGGACCTCGTTGACACCTGGAAGAGCTTCAAACCGAGATACAAGCACTCC
CACTTCATCTGGAACCGCACGGAACTCCTGACCCTTGACCCCCACAATGTGGACTACCTATTGGGTTTCTTCGA
GCCAGGGGACATGCAGTACGAGCTGAACAGGAACAACGTGACGGACCCGTCACTCTCCGAGATGGTGGTGGT
GGCCATCCAGATCCTGCGGAAGAACCCCAAAGGCTTCTTCTTGCTGGTGGAAGGAGGCAGAATTGACCACGG
GCACCATGAAGGAAAAGCCAAGCAGGCCCTGCATGAGGCGGTGGAGATGGACCGGGCCATCGGCCACGCAG
GCAGCTTGACCTCCTCGGAAGACACTCTGACCGTGGTCACTGCGGACCATTCCCACGTCTTCACATTTGGTGGA
TACACCCCCCGTGGCAACTCTATCTTTGGTCTGGCCCCCATGCTGAGTGACACAGACAAGAAGCCCTTCACTGC
CATCCTGTATGGCAATGGGCCTGGCTACAAGGTGGTGGGCGGTGAACGAGAGAATGTCTCCATGGTGGACTA
TGCTCACAACAACTACCAGGCGCAGTCTCCTGTGCCCCTGCGCCACGAGACCCACGGCGGGGAGGACGTGGC
CGTCTTCTCCAAGGGCCCCATGGCGCACCTGCTGCACGGCGTCCACGAGCAGAACTACGTCCCCCACGTGATG
GCGTATGCAGCCTGCATCGGGGCCAACCTCGGCCACTGTGCTCCTGCCAGCTCGGCAGGCAGCCTTGCTGCAG
GCCCCCTGCTGCTCGCTCTGGCCCTCTACCCCCTGAGCGTCCTGTTCTGA'

Ensuite, elle va être traduite en protéine par le logiciel développé comme suit :

Met Ile Ile Ser Pro Phe Leu Val Leu Ala Ile Gly Thr Cys Leu Thr Asn Ser Leu Val Pro Glu Lys Glu Lys Asp
Pro Lys Tyr Trp Arg Asp Gln Ala Gln Glu Thr Leu Lys Tyr Ala Leu Glu Leu Gln Lys Leu Asn Thr Asn Val Al
Lys Asn Val Ile Met Phe Leu Gly Asp Gly Met Gly Val Ser Thr Val Thr Ala Ala Arg Ile Leu Lys Gly Gln Leu
His His Asn Pro Gly Glu Glu Thr Arg Leu Glu Met Asp Lys Phe Pro Phe Val Ala Leu Ser Lys Thr Tyr Asn
Thr Asn Ala Gln Val Pro Asp Ser Ala Gyl Thr Ala Thr Ala Tyr Leu Cys Gly Val Lys Ala Asn Glu Gly Thr Val
Gly Val Ser Ala Ala Thr Glu Arg Ser Arg Cys Asn Thr Thr Gln Gly Asn Glu Val Thr Ser Ile Leu Arg Trp Ala
Lys Asp Ala Gly Lys Ser Val Gly Ile Val Thr Thr Thr Arg Val Asn His Ala Thr Pro SerAla Ala Tyr Ala His Ser
Ala Asp Arg Asp Trp Tyr Ser Asp Asn Glu Met Pro Pro Glu Ala Leu Ser Gln Gly Cys Lys Asp Ile Ala Tyr
Gln Leu Met His Asn Ile Arg Asp Ile Asp Val Ile Met Gly Gly Gly Arg Lys Tyr Met Tyr Pro Lys Asn Lys Thr
Asp Val Glu Tyr Glu Ser Asp Glu Lys Ala Arg Gly Thr Arg Leu Asp Gly Leu Asp Leu Val Asp Thr Trp Lys
Ser Phe Lys Pro Arg Tyr lys His Ser His Phe Ile Trp Asn Arg Thr Glu Leu Leu Thr Leu Asp Pro His Asn Val
Asp Tyr Leu Leu Gly Phe Phe Glu Pro Gly Asp Met Gln Tyr Glu Leu Asn Arg Asn Asn Val thr Asp Pro Ser
Leu Ser Glu Met Val Val Val Ala Ile Gln Ile leu Arg Lys Asn Pro lys Gly Phe Phe Leu Leu Val Glu Gly Gly
Arg Ile Asp His Gly His His Glu Gly Lys Ala Lys Gln Ala Leu His Glu Ala Val Glu Met Asp Arg Ala Ile Gly
His Ala Gly Ser Leu Thr Ser Ser Glu Asp Thr Leu Thr Val Val Thr Ala Asp his Ser His Val Phe Thr Phe Gly
Gly Tyr Thr Pro Arg Gly Asn Ser Ile Phe Gly Leu Ala Pro Met Leu Ser Asp Thr Asp Lys Lys Pro Phe Thr
Ala Ile Leu Tyr Gly Asn Gly Pro Gly Tyr Lys Val Val Gly Gly Glu Arg Glu Asn Val Ser Met Val Asp Tyr Ala
His Asn Asn Tyr Gln Ala Gln Ser Pro Val Pro Leu Arg His Glu Thr His Glu Gly Glu Asp val Ala Val phe Ser
Lys Gly Pro Met Ala His Leu Leu His Gly Val His Glu Gln Asn Tyr Val Pro His Val Met Ala Tyr Ala Ala Cys
Ile Gly Ala Asn Leu Gly His Cys Ala Pro Ala Ser Ser Ala Gly Ser Leu Ala Ala Gly Pro Leu Leu Leu Ala Leu
Ala Leu tyr Pro Leu ser Val leu Phe Stop

Figure 22. La séquence protéique, qui représente la traduction issue de la séquence de l'Alkalin par le logiciel, en séquence peptidique en trois lettres.

Figure 23. La séquence protéique, qui représente la traduction issue de la séquence d'Alkalin par le logiciel, en séquence peptidique en 1 symbole.

D'autre part, nous prenons la séquence protéique qui représente la traduction de la partie codante de la séquence Alkalin phosphatase et qui se trouve dans la banque EMBL. La figure suivante représente la séquence de cette protéine.

```
MISPFLVLAIGTCLTNSLVPEKEKDPKYWRDQAQETLKYALELQKLNTNVAKNVIMFLGDGMGVSTVTAARILKGQLHHN
PGEETRLEMDKFPFVALSKTYNTNAQVPDSAGTATAYLCGVKANEGTVGVSAATERSRCNTTQGNEVTSILRWAKDAGKS
VGIVTTTRVNHATPSAAYAHSADRDWYSDNEMPPEALSQGCKDIAYQLMHNIRDIDVIMGGGRKYMYPKNKTDVEYESDE
KARGTRLDGLDLVDTWKSFKPRYKHSHFIWNRTELLTLDPHNVDYLLGFFEPGDMQYELNRNNVTDPSLSEMVVVAIQIL
RKNPKGFFLLVEGGRIDHGHHEGKAKQALHEAVEMDRAIGHAGSLTSSEDTLTVVTADHSHVFTFGGYTPRGNSIFGLAP
MLSDTDKKPFTAILYGNGPGYKVVGGERENVSMVDYAHNNYQAQSPVPLRHETHGGEDVAVFSKGPMAHLLHGVHEQNYV
PHVMAYAACIGANLGHCAPASSAGSLAAGPLLLALALYPLSVLF
```

Figure 24. La séquence protéique, qui représente la traduction issue de la séquence de l'Alkalin par le logiciel, d'après la banque de données EMBL.

Enfin, nous avons comparé la séquence protéique qui se trouve dans la banque EMBL avec celle que nous avons obtenue à partir du logiciel développé en utilisant le logiciel d'alignement CLUSTAL OMEGA.

La figure suivante représente l'interface du logiciel Clustal Omega.

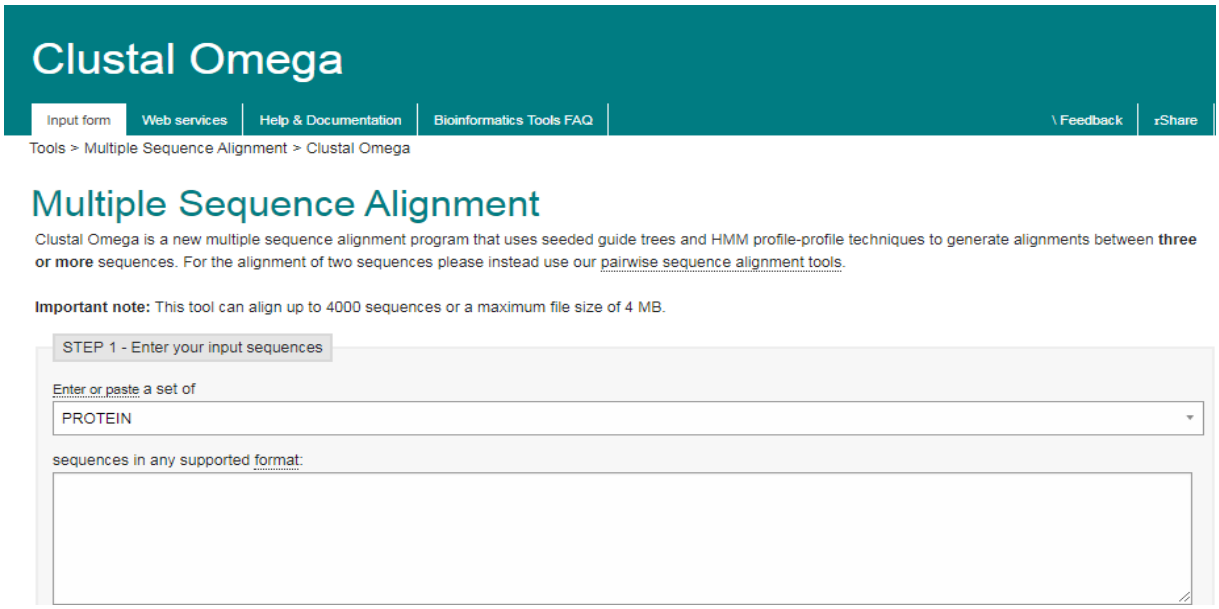


Figure 25. L'interface du logiciel Clustal Omega.

Nous obtenons le résultat suivant :

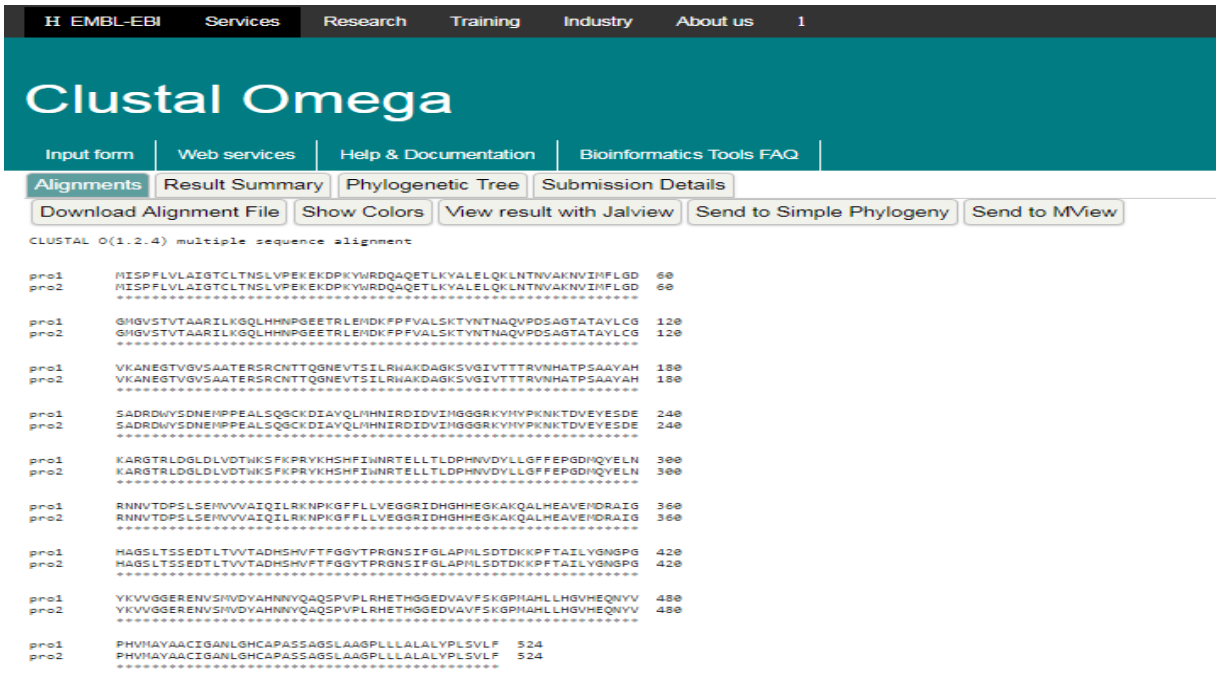


Figure 26. Le résultat du Clustal Omega qui indique l'identité entre les deux protéines.

Nous remarquons qu'il existe une ressemblance parfaite (à 100%). Ceci confirme que la séquence protéique qui se trouve dans la banque EMBL est la même séquence que nous avons obtenue à partir du logiciel développé.

Finalement, nous avons confirmé que notre logiciel donne des résultats corrects.

1.2. La validation

La validation est une activité qui vérifie que la conception du produit satisfait à l'usage auquel il est destiné (le logiciel doit faire ce dont l'utilisateur a besoin).

Notre objectif est de construire un logiciel capable de calculer la traduction des nouvelles séquences ADN qui ne sont pas encore découvertes.

Dans la biologie, une séquence imaginaire est dite chimère. Cette séquence est construite comme suit :

Premièrement, nous construisons à partir d'une combinaison des bases (A, T, G et C) une séquence qui contient un nombre de bases supérieur ou égal à 300 bases.

Deuxièmement, nous vérifions que cette séquence n'est pas encore découverte. Pour ce faire, nous vérifions que cette séquence ne se trouve pas dans les banques des données (elle n'est pas identifiée) et il n'existe pas de ressemblances entre cette séquence et les séquences existantes.

La séquence suivante représente un exemple d'une séquence chimères.

```
ATGGTATATGTACGTCCCGTCACGACTCTAACTCGTGTCATCGTCAGTCGTCAGTTGTCGATGCGTAGCTGCAC
GTGCAGTACGATGACTACGTATTGCTACATGATGCGTATGTGCTACAAACCCGGGTTTACGTGCATGATGGGC
CCTATCATGCGACGTACGACGACGGCAGCAGTCGTCGCAAAATCATACAAAGCAGGGCTATACTCACGTGCTC
GACCGGGTACAATCTGCTGGGACAAGAATTTCCCTGCGACGGCAATTTTACCGGCGGGGAAACCCGTAATGC
GACGTAGTAAATTTGGGCCCCCGGGCTTGGATGTAGTGCCGCGCAGCCGACGTGCTGTTATGAAGAATAACG
ATGCAACAGCGTACTGTATGGCCTACTGGGCCTACTGTAGTGCAAAATGCGCGATATTACGCTGCCGCAGATT
CAAATTTGGGCCCTACCGAATGGTGGACTGGGGCCCGTACTCCTTGGCTGGGTGGGTTATCGCGCGGAGCAG
TTCCTTCGACGGGGCACCCTTTCCCAAATGTAATAAATTTGTCGTCCAAAGTACGGCACAGCAGCAGCCGACGC
ATGTTTTTCTGTTGCGCTGCCGCAGCGAACAATGGCGATCCCCCTCCACCGCCACAGCGATCAGTCGTTGTAGG
GGTCACGGCAATGCGAAAACCCGGGGCAAACGCAGCAGCAGCAGCCCGCAGCAGCAAACGACGCAGCAGCA
GTGA
```

Figure 27. Exemple d'une séquence chimère.

Nous introduisons la séquence chimère dans le programme BLAST (Basic local alignement Search tool). Ce dernier nous permet de retrouver rapidement dans des bases de données, les séquences répertoriées ayant des zones de similitude avec cette séquence chimère.

Nous choisissons de blaster notre séquence avec des séquences du gène codant l'ARN16S bactérien.

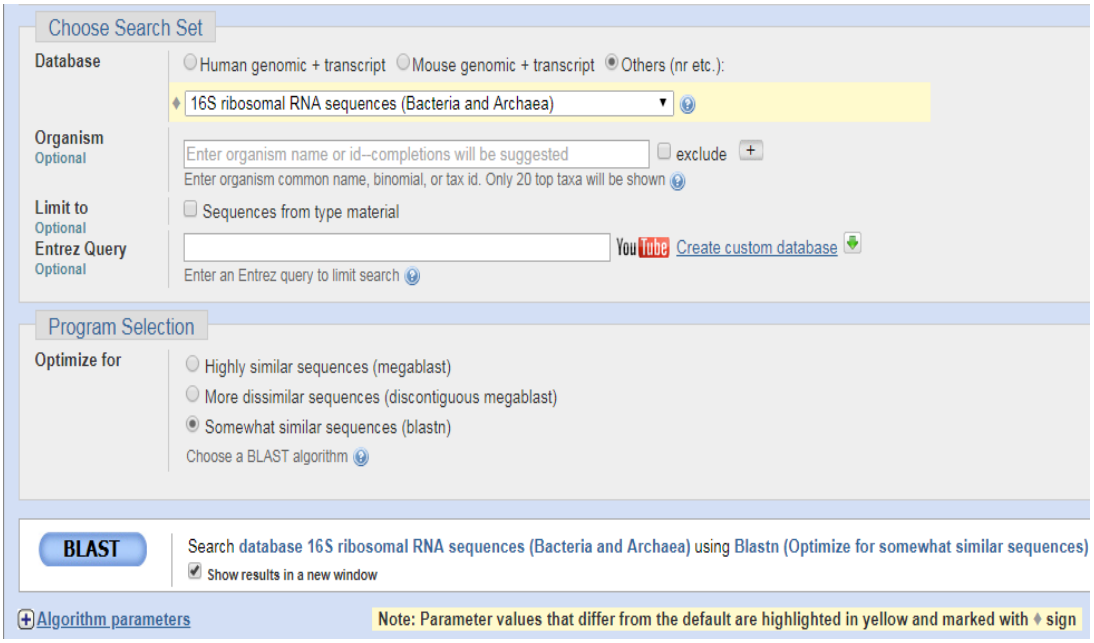


Figure 28. L'interface du logiciel BLAST.

Nous obtenons le résultat suivant :

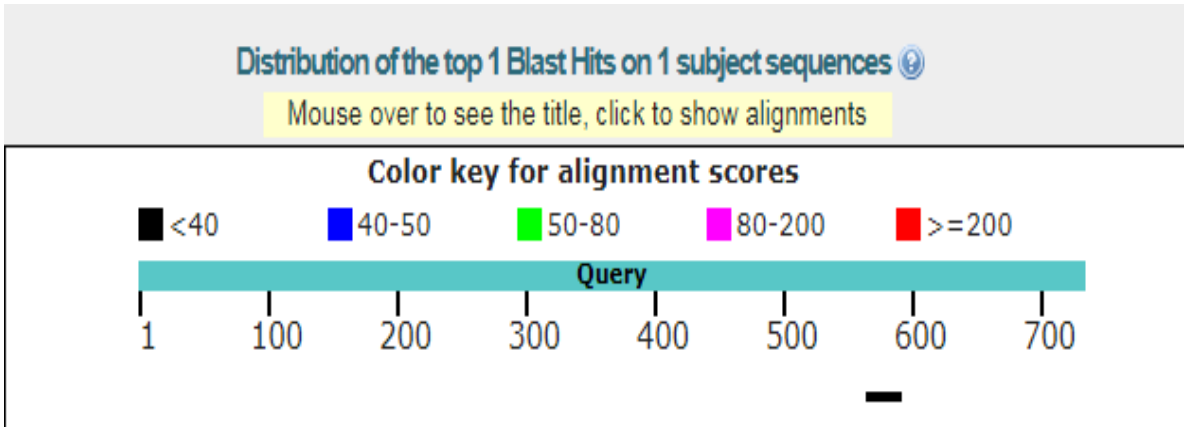


Figure 29. Illustration représente le résumé graphique de la séquence chimère dans le logiciel BLAST.

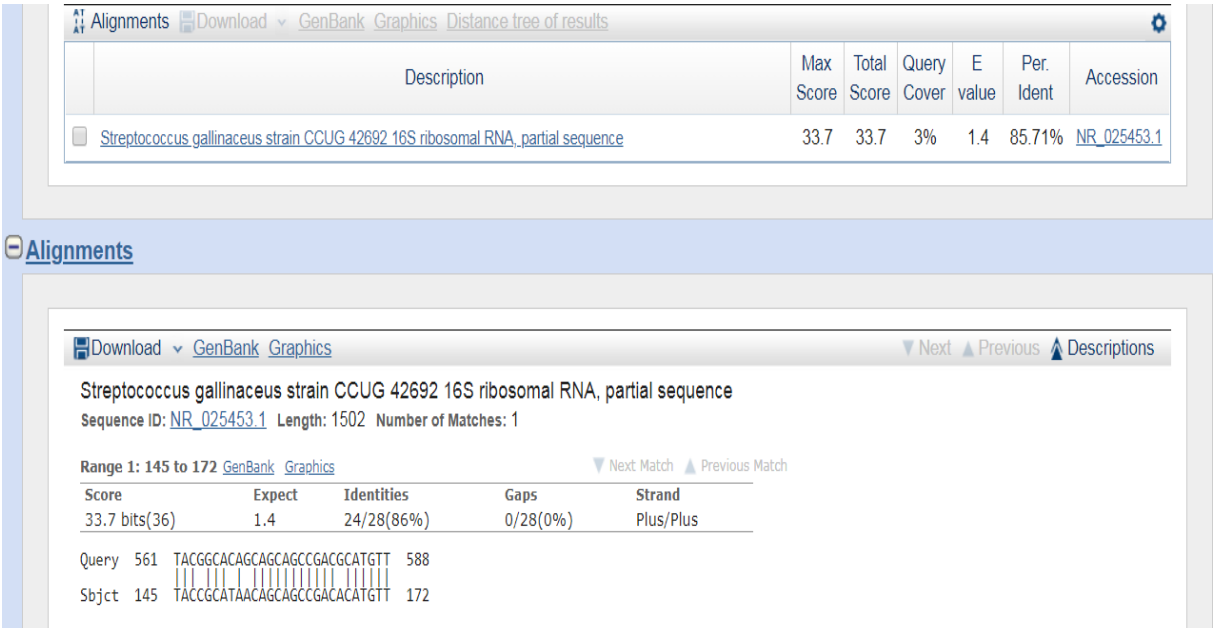


Figure 30. Illustration qui représente le pourcentage de l'identité de la séquence chimère avec les séquences naturelles dans le logiciel BLAST.

Alors, le pourcentage 85,71% nous montre que la séquence chimère est très éloignée de *streptococcus qallinaceus strain*. Donc, elle ne ressemble à aucune séquence dans les bases des données.

Par conséquent, nous confirmons que cette séquence est une séquence chimère qui n'existe encore pas parmi les séquences répertoriées.

Troisièmement, nous exécutons le logiciel développé sur cette séquence. La séquence protéique suivante représente la traduction de la séquence chimère vue précédemment en utilisant le logiciel développé.

Met Val Tyr Val Arg Pro Val Thr Thr Leu Thr Arg Val Ile Val Ser Arg Gln Leu Ser Met
Arg Ser Cys Thr Cys Ser Thr Met Thr Tyr Cys Tyr Met Met Arg Met Cys Tyr Lys Pro
Gly Phe Thr Cys Met Met Gly Pro Ile Met Arg Arg Thr Thr Thr Ala Ala Val Val Ala Lys
Ser Tyr Lys Ala Gly Leu Tyr Ser Arg Ala Arg Pro Gly Thr Ile Cys Trp Asp Lys Asn Phe
Pro Ala Thr Ala Ile Leu Pro Ala Gly Lys Pro Val Met Arg Arg Ser Lys Phe Gly Pro Pro
Gly Leu Asp Val Val Pro Arg Ser Arg Arg Ala Val Met Lys Asn Asn Asp Ala Thr Ala Tyr
Cys Met Ala Tyr Trp Ala Tyr Cys Ser Ala Lys Cys Ala Ile Leu Arg Cys Arg Arg Phe Lys
Phe Gly Pro Tyr Arg Met Val Asp Trp Gly Pro Tyr Ser Leu Ala Gly Trp Val Ile Ala Arg
Ser Ser Ser Phe Asp Gly Ala Pro Phe Pro Lys Cys Asn Lys Phe Val Val Gln Ser Thr Ala
Gln Gln Gln Pro Thr His Val Phe Leu Leu Arg Cys Arg Ser Glu Gln Trp Arg Ser Pro Ser
Thr Ala Thr Ala Ile Ser Arg Cys Arg Gly His Gly Asn Ala Lys Thr Arg Gly Lys Arg Ser Ser
Ser Ser Pro Gln Gln Gln Thr Thr Gln Gln Gln Stop

Figure 31. Séquence protéique représente la traduction de la séquence chimère par le logiciel développé en trois lettres.

la séquence pro codée en 1 symbole

v4 =

'MVYVRPVTTLTRVIVSRQLSMRSCTCSTMTTYCYMMRMCYKPGFTCMMGPIMRRTTTAAVVAKSYKAGLYSRA
RPGTICWDKNFPATAILPAGKPVMRRSKFGPPGLDVVPRSRRAVMKNNDATAYCMAYWAYCSAKCAILRCRRFKF
GPYRMVDWGPYSLAGWVIARSSSFDGAPFPKCNKFVVQSTAQQQPTHVFLLRCRSEQWRSPSTATAISRCRGHG
NAKTRGKRSSSSPQQQTTQQQ'

Figure 32. Séquence protéique qui représente la traduction de la séquence chimère par le logiciel en un seul symbole.

Par ailleurs, nous introduisons cette séquence protéique dans le programme BLAST pour l'identifier.

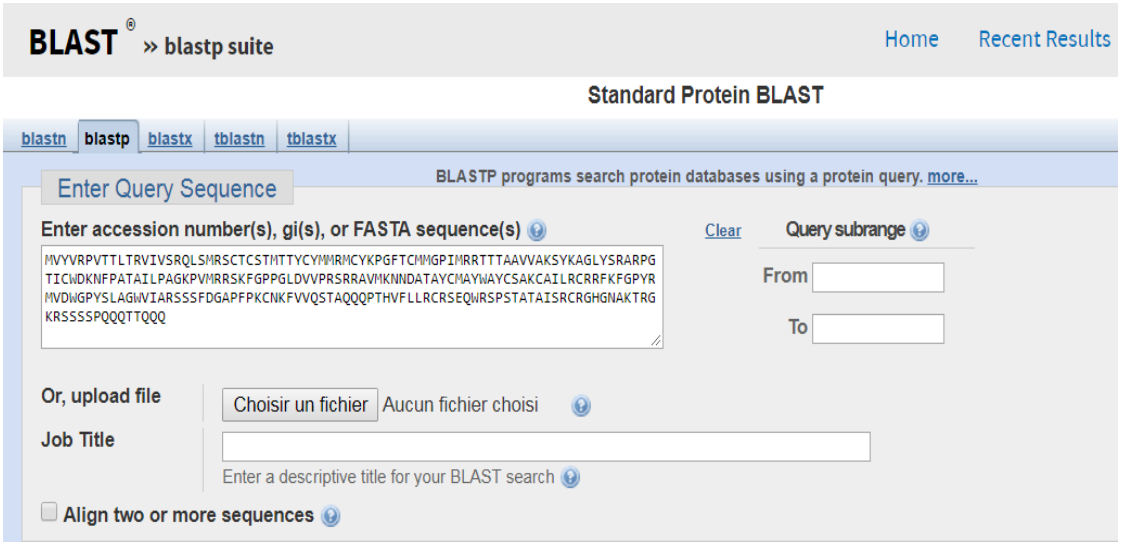


Figure 33. Illustration de la séquence protéique dans le programme BLAST.

Nous obtenons le résultat suivant :

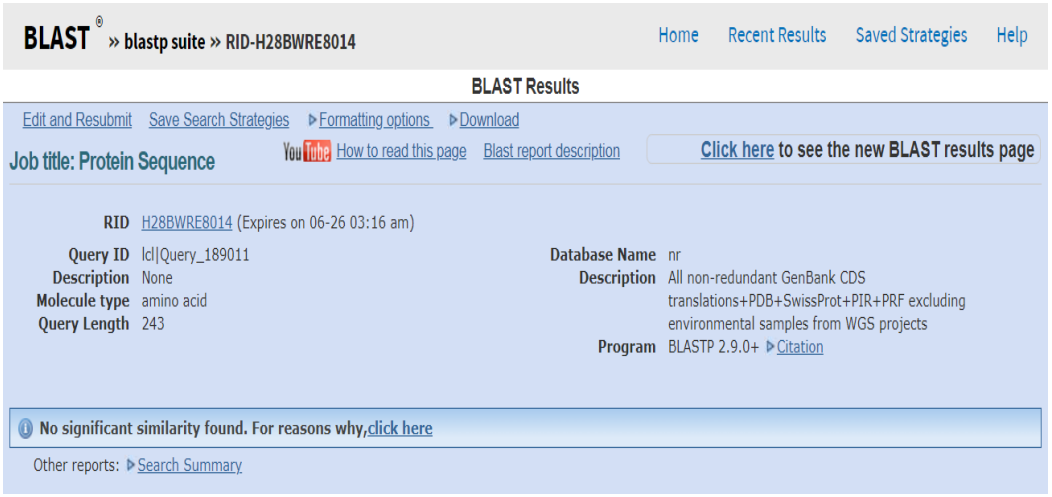


Figure 34. Le résultat de ressemblance de la séquence protéique dans le programme BLAST.

Ce résultat nous confirme que le logiciel développé est un logiciel qui permet de traduire des séquences même si elles n'existent pas naturellement et même si elles n'avaient pas déjà été. Donc, c'est pourquoi même quand les banques de données sont incapables de nous donner une issue, le logiciel nous donne une idée sur la protéine qui peut être construite par une séquence chimère.

2. Comparaison avec les logiciels existants

Il existe des logiciels qui font la traduction des séquences nucléiques en protéines. Selon la littérature ce, le logiciel BLAST, le logiciel Anagène et le logiciel développé par Mehdi et Meziani (2018) représentent le sujet de comparaison avec le logiciel que nous avons développé. :

2.1. BLAST

BLAST fait la traduction de façon indirecte où nous pouvons introduire une séquence naturelle (existe dans les banques des données). Il fait la recherche dans les banques des données pour trouver une séquence qui se ressemble à la séquence introduite. Ensuite, il va extraire la traduction à partir de la fiche descriptive de la séquence introduite.

Par contre, quand on a introduit une séquence chimère (n'existe pas dans les banques des données), il ne donne aucun résultat.

Alors, nous confirmons que le principe de fonctionnement du BLAST n'est pas la modélisation (ne simule pas le processus naturel). Par contre, le principe de fonctionnement de notre logiciel est la modélisation qui simule le principe de fonctionnement de traduction naturelle des séquences d'ADN en protéines : nous introduisons la séquence d'ADN. Cette dernière va être traduite en protéines sans passer par la recherche sur les banques des données.

2.2. Anagène

Anagène est un logiciel d'analyse de séquences nucléiques et protéiques. Il comporte une banque de données et différents outils d'analyse et de traitement des données. Parmi les traitements qu'Anagène peut effectuer, la traduction de l'ADN en protéines. Ce programme nécessite l'intervention de l'utilisateur à chaque étape de la traduction. Il demande à l'utilisateur d'introduire la séquence ADN pour faire la transcription. Puis, il demande à l'utilisateur d'introduire la séquence ARN messager pour faire la traduction. Nous remarquons deux défauts dans ce logiciel. Le premier est qu'il nécessite l'intervention de l'utilisateur à chaque étape. Le deuxième est qu'il n'effectue pas l'étape de la maturation (à partir de l'ARN messager, il fait la traduction). Alors, l'utilisateur peut introduire une séquence d'ARN messager où le nombre de bases n'est pas un diviseur de 3. Dans ce cas, la traduction sera impossible.

Par contre, le logiciel développé ne nécessite pas l'intervention de l'utilisateur à chaque étape. La traduction se fait automatiquement. D'autre part, le logiciel développé effectue l'étape de la maturation.

2.3. Logiciel développé par Mehdi et Meziani (2018)

Mehdi et Meziani (2018) ont développé un logiciel qui simule la traduction automatique d'ADN en protéines. Dans ce qui suit, nous discutons des améliorations que nous avons apportées sur ce logiciel.

Premièrement, ils ont vérifié le logiciel par l'exploitation des séquences naturelles et issues des banques de données. Nous remarquons qu'ils ont fait sortir la région codante depuis la fiche descriptive de la séquence complète. De ce fait, elles ont trouvé la région codante manuellement. Elles ont détecté la base où commence le codon start et la base où se termine le codon stop comme dans la figure suivante :

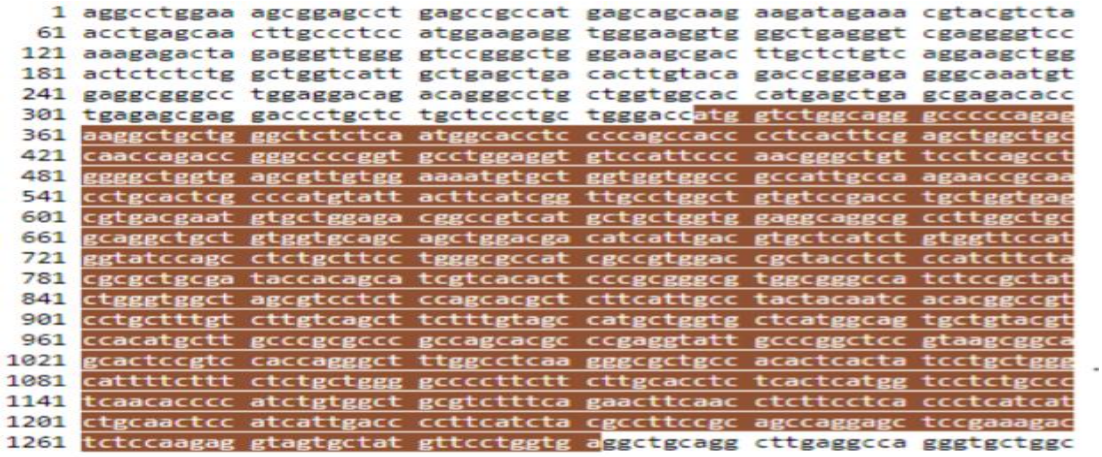


Figure 35. Représentation de la région codante à partir d'une séquence d'ADN complète (Mehdi et Meziani, 2018).

Malgré qu'elle existe explicitement la partie codante dans la fiche descriptive comme dans la figure suivante.

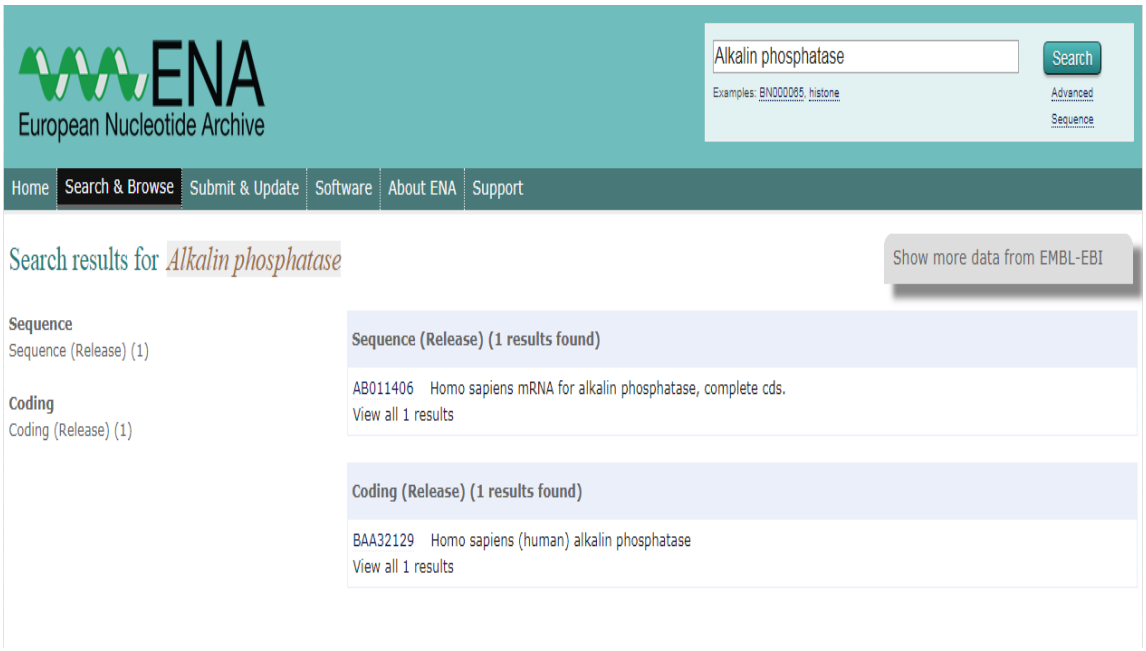


Figure 36. Fiche desciptive de l'EMBL qui représente la région codante de Alkain Phosphatase.

la façon de prendre la région codante (manuelle) provoque beaucoup de risques. A titre d'exemple, le risque de faire des erreurs (comme prendre des bases en plus) et qui n'appartient pas à la partie codante ou bien obtenir une partie codante incomplète.

Par contre, dans notre travail, nous extraiyons la partie codante à partir de la fiche descriptive de la coding sequence et pas à partir de la fiche descriptive de la séquence elle-même.

Deuxièment, le logiciel développé par **Mehdi et Meziani (2018)** permet d'afficher la protéine avec le code en 3 lettres de chaque acide aminé. Comme dans la figure suivante :

Met Ile Ile Ser Pro Phe Leu Val Leu Ala Ile Gly Thr Cys Leu Thr Asn Ser Leu Val Pro Glu Lys Glu Lys Asp
Pro Lys Tyr Trp Arg Asp Gln Ala Gln Glu Thr Leu Lys Tyr Ala Leu Glu Leu Gln Lys Leu Asn Thr Asn Val Al
Lys Asn Val Ile Met Phe Leu Gly Asp Gly Met Gly Val Ser Thr Val Thr Ala Ala Arg Ile Leu Lys Gly Gln Leu
His His Asn Pro Gly Glu Glu Thr Arg Leu Glu Met Asp Lys Phe Pro Phe Val Ala Leu Ser Lys Thr Tyr Asn
Thr Asn Ala Gln Val Pro Asp Ser Ala Gyl Thr Ala Thr Ala Tyr Leu Cys Gly Val Lys Ala Asn Glu Gly Thr Val
Gly Val Ser Ala Ala Thr Glu Arg Ser Arg Cys Asn Thr Thr Gln Gly Asn Glu Val Thr Ser Ile Leu Arg Trp Ala
Lys Asp Ala Gly Lys Ser Val Gly Ile Val Thr Thr Thr Arg Val Asn His Ala Thr Pro SerAla Ala Tyr Ala His Ser
Ala Asp Arg Asp Trp Tyr Ser Asp Asn Glu Met Pro Pro Glu Ala Leu Ser Gln Gly Cys Lys Asp Ile Ala Tyr
Gln Leu Met His Asn Ile Arg Asp Ile Asp Val Ile Met Gly Gly Gly Arg Lys Tyr Met Tyr Pro Lys Asn Lys Thr
Asp Val Glu Tyr Glu Ser Asp Glu Lys Ala Arg Gly Thr Arg Leu Asp Gly Leu Asp Leu Val Asp Thr Trp Lys
Ser Phe Lys Pro Arg Tyr lys His Ser His Phe Ile Trp Asn Arg Thr Glu Leu Leu Thr Leu Asp Pro His Asn Val
Asp Tyr Leu Leu Gly Phe Phe Glu Pro Gly Asp Met Gln Tyr Glu Leu Asn Arg Asn Asn Val thr Asp Pro Ser
Leu Ser Glu Met Val Val Val Ala Ile Gln Ile leu Arg Lys Asn Pro lys Gly Phe Phe Leu Leu Val Glu Gly Gly
Arg Ile Asp His Gly His His Glu Gly Lys Ala Lys Gln Ala Leu His Glu Ala Val Glu Met Asp Arg Ala Ile Gly
His Ala Gly Ser Leu Thr Ser Ser Glu Asp Thr Leu Thr Val Val Thr Ala Asp his Ser His Val Phe Thr Phe Gly
Gly Tyr Thr Pro Arg Gly Asn Ser Ile Phe Gly Leu Ala Pro Met Leu Ser Asp Thr Asp Lys Lys Pro Phe Thr
Ala Ile Leu Tyr Gly Asn Gly Pro Gly Tyr Lys Val Val Gly Gly Glu Arg Glu Asn Val Ser Met Val Asp Tyr Ala
His Asn Asn Tyr Gln Ala Gln Ser Pro Val Pro Leu Arg His Glu Thr His Glu Gly Glu Asp val Ala Val phe Ser
Lys Gly Pro Met Ala His Leu Leu His Gly Val His Glu Gln Asn Tyr Val Pro His Val Met Ala Tyr Ala Ala Cys
Ile Gly Ala Asn Leu Gly His Cys Ala Pro Ala Ser Ser Ala Gly Ser Leu Ala Ala Gly Pro Leu Leu Leu Ala Leu
Ala Leu tyr Pro Leu ser Val leu Phe Stop

Figure 37 . Modélisation de la séquence d'ADN (alkain phosphatase) en protéine par le logiciel développé par (Mehdi et Meziani, 2018).

Mais, dans les banques de données les protéines sont représentées avec le code en une seule lettre pour chaque acide aminé. De ce fait, comment elle a effectué l'alignement entre la séquence protéique obtenue par le logiciel et la protéine issue de la banque pour vérifier que le logiciel donne des résultats non erronés. Donc, nous constatons, l'alignement a été fait manuellement.

Par contre, le logiciel que nous avons développé permet d'afficher la protéine en deux modes :

1- avec le code en 3 lettres de chaque acide aminé afin que la protéine devienne lisible pour l'utilisateur humain.
2- avec le code en une seule lettre pour acide aminé afin que la protéine devienne exploitable par les logiciels tels que les logiciels d'alignement et elle devient compatible avec l'affichage des séquences enregistrées dans les banques de données.

D'autre part, pour vérifier que la protéine obtenue par le logiciel, qui représente la traduction d'une séquence ADN naturelle, est la même protéine de cette séquence ADN, qui est issue de la fiche descriptive de cette séquence ADN, nous avons exploité le logiciel d'alignement Clustal Omega.

Troisièment, **Mehdi et Meziani (2018)** n'ont pas exécuté leur logiciel sur une séquence chimère. Par contre, nous avons testé le logiciel développé sur des séquences chimères afin de prouver la capacité de notre logiciel de donner les résultats prévus pour les séquences qui ne sont pas découvertes encore.

Quatrièment, **Mehdi et Meziani (2018)** ont laissé l'utilisateur d'introduire une séquence ADN qui ne satisfait pas les conditions d'une partie codante. Et comme le logiciel n'est pas intelligent, il peut donner des résultats. Cependant, ces résultats n'ont pas de sens, car la séquence ADN ne représente pas une partie codante sollicitée d'être interprétée.

Cependant, dans notre logiciel, la séquence ADN doit vérifier les conditions d'une partie codante et qui sont :

- La taille de cette séquence doit être supérieure à 300 Pb
- Les trois premières bases de cette séquence sont A, U, G
- Le nombre des bases de cette séquence est un nombre diviseur du nombre 3
- Les trois dernières bases de cette séquence sont soit T, A, A ou T, A, G ou T, G, A.
- Les triplets (T, A, A), (T, A, G) et(T, G, A) ne figurent pas au milieu de la séquence.

Avec ces conditions, nous garantirons que cette séquence même elle n'est pas découverte encore, mais elle peut être découverte dans le futur. Ou bien, qu'elle donne une séquence protéique, qui représente une enzyme salubre ou un médicament, alors on peut produire cette séquence ADN dans le laboratoire (*in vitro*).

Finalement, nous résumons le discours précédent sur la comparaison entre le logiciel que nous avons développé et celui développé par **Mehdi et Meziani (2018)** de la façon suivante :

- ✓ Nous avons corrigé la vérification de leur logiciel.
- ✓ Nous avons corrigé la sortie (la façon d'afficher la séquence protéine)
- ✓ Nous avons convergé vers la réalité par l'introduction de la séquence chimère.

Conclusion générale

La bioinformatique est une discipline qui vise le traitement automatique de l'information biologique.

Dans le cadre de ce travail de master, nous avons traité un problème très important en bioinformatique celui de la traduction automatique des séquences d'ADN en protéines. C'est la raison pour laquelle nous avons développé un logiciel qui permet de traduire des séquences d'ADN naturelles et des séquences d'ADN qui n'existent pas dans la nature en protéines.

Nous avons aussi présenté dans ce mémoire des logiciels qui permettent d'aligner et de comparer plusieurs séquences nucléiques et protéiques. L'utilisation de ces logiciels nous a aidés à confirmer que notre logiciel donne des résultats corrects.

Nous avons construit des séquences chimères et à l'aide des logiciels d'alignement, nous avons garanti que ces séquences ne sont pas encore découvertes.

Les résultats obtenus par l'exécution du logiciel développé nous ont confirmé qu'il permet de traduire des séquences même si elles n'existent pas naturellement et même si elles n'avaient pas été déjà séquencées (parce qu'elles peuvent exister, mais elles ne sont pas encore séquencées). Donc, c'est pourquoi même quand les banques de données sont incapables de nous donner une issue, ce logiciel nous donne une idée sur la protéine qui peut être construite par une séquence chimère. La protéine qui est issue d'une séquence chimère peut-être une enzyme salubre ou un médicament. Alors, le logiciel développé peut nous guider à produire les séquences ADN dans le laboratoire (*in vitro*) qui permet de produire des enzymes salubres ou des médicaments. De ce fait, ce logiciel peut être exploité dans les domaines : pharmaceutique, cosmétique et industrie.

Enfin, après plusieurs comparaisons avec les logiciels qui font la traduction des séquences ADN en séquences protéiques, on peut dire que notre logiciel possède la capacité de simuler le processus naturel de la traduction des séquences d'ADN en protéines.

Comme perspectives, nous visons à compléter le logiciel de telle sorte, il devient capable de détecter la partie codante à partir d'une séquence d'ADN complète. Nous visons aussi à améliorer le logiciel afin de simuler la génération des protéines produites par une combinaison des gènes.

Références bibliographiques

Références bibliographiques

Académie française dictionnaire, 9éme édition, 1966.

Ameziane, N., Bogard, M., Lamoril, J (2006). «Principes de biologie moléculaire en biologie clinique». Paris : Dragos Bobu. 705p.

Beroud, C. (2010). Base de donnés et outils Bioinformatique outils en génétique. Support de cours : collège National des enseignants et praticiens de génétique médicale virtuelle Francophone, 19p.

Boujard, D., Anselme, B., cullin, C. et Ragnénès-Nicol, C (2012). «Biologie cellulaire et moléculaire». Paris : Dunod. 489p.

Breton, P. (1990). « Une histoire de l'informatique ». Paris, Edition du seuil.

Caffiau, S. (2006). Approche dirigée par les modèles pour la conception et la validation des applications internatives : une démarche basée sur la modélisation des taches. Thèse pour l'obtention du doctorat, Ecole Nationale Supérieure De Mécanique et D'aérotechnique.132p.

Cherfa, D., Chibouh, F. (2014). Système d'information et son rôle au sein d'Entreprise. Mémoire Master Recherche : Sociologie du travail et des ressources humaines. Bejaïa : Université Abderrahmane mira-Bejaïa, Faculté des sciences humaine et sociales, Département des sciences sociales, 75p.

Chibouti, F., Cherfa, D. (2014). Système d'information et son rôle au sein de l'entreprise. Mémoire de master : Sociologie du travail des ressources humaines, Université Abderrahmane mira-Bejaïa. Faculté des sciences humaines et sociales, Département des sciences sociales, 75p.

Collet, P. 2012. Spécification du logiciel-OCL. Licence 3. Parcours informatique et MIAGE. Université Nice Sophia Antipolis. 132p.

Delmas, Y. (2009). « Histoire de l'informatique, d'internet st du web ». [En ligne] : http://delmasrigoutsos.nom.fr/document/YDelmas-histoire-informatique/

Duboz, R., Ramat, E., et Quesnel, G. (2004). Système multi-agents et théorie de la modélisation et de la simulation : Une analogie opérationnelle.

Duboz, R., Ramat, E., et Quesnel, G., Garcia, F. 2006. Théorie de la modélisation et de la simulation. Support de cours : Institue nationale de la recherche agronomique, Laboratoire d'informatique du Littoral, 162P.

Dictionnaire d'informatique, M. GINGUAY, A. LAURET, Masson, 4° édition, (1990).

Dictionnaire de l'informatique, sous la direction de Pierre Morvan, Larousse, 1981.

Girond, F. 2011. Alignement de séquences. Support de cours : Observation à l'aide de l'outil graphique : La dot plot, La bioinformatique : LATRONCHECEDEX- France : Laboratoire TIMC-IMAG/Equipe Dyctim, 54p.

Housset, C., Raisonnier. (2006). Biologie moléculaire. Biochimie PCEM1 Université Paris-VI. 204p.

Mondjo, L. (1961). Cycle de vie d'un logiciel. Sup info international University. Ecole d'informatique de paris, Leader en France-La grande. Ecole de l'informatique, du numérique et du management Fondé en 1965, Publier le 27/10/2009

https://www.supinfo.com/articles:sing 3210-Cycle de vie logiciel.

Layeb, A. (2005). Approche quantique évolutionnaire pour l'alignement multiple de séquence bioinformatique. Mémoire de magister : information et computation. Faculté : science de l'ingénieur, Université Mentouri Constantine. Département de l'informatique, 136p.

Legrand, S. 2016. Banque de données de séquences. Support de cours : de l'équipe Bonsaï, CRISTAL UMR 9189 : Science et technologies. Lille-France : Université de Lille, 56p.

Lodish, B., Matsudaira, K., Krieger, K., Zipursky, S .2003. Molecular Cell Biology. 6th Ed. 937p.

Longuet, D. 2017. Introduction au génie logiciel et à la modélisation. Support de cours : Polytech Paris-Sud, Formation initiale 3° Année : Spécialité informatique, 52p.

Luchetta, P. (2009). «Biologie moléculaire en 30 fichiers 2°éd». Paris : Dunod. 155p.

Mahec, G. (2008). Gestion des bases de données biologiques sur grilles de calcule. Thèse de doctorat : Informatique. Franc : Equipe d'accueil : Pcsv, In2p3, clermont-Ferraud, Ecole doctorale des sciences pour ingénieur, Université Blaire pascal, 175p.

Makowski, F.1981, Traitement automatique de données. Paris [En ligne]. (Page consulté le 16/6/2019).http://www.marche-public.fr/Terminologie/Entrees/traitement-automatique-de-donnees.htm1.

Mehdi, Z., Meziani F., (2018). Automatisation du traducteur des séquences ADN en protéines. Mémoire de master : Mycologie et biotechnologie fongique. Filière : sciences biologiques, Université du frère mentouri constantine1. Faculté des sciences de la nature et de la vie, 75p.

Mesguich, A., Normier, B. (1982). «Comprendre les bases de données ». Bulletin des bibliothèques de France (BBF) n°6, 379-379.

Meshoul, S. (2007). Optimisation Multi-objectif pour l'Alignement Multiples de séquences. Mémoire présenté en vue de l'obtention du diplôme de magistère en informatique : Génie logiciel et Intelligence artificielle Constantine –Algérie : Université mentouri Constantine ,134p.

Mezhoud, K. (2004). Alignement de séquences principes et méthodes. Cour : Ir. Agronome phD. Toxicologie, protonique, Bioinformatique.sidi Thabet-Tunis : centre national des sciences et technologies nucléaires, 69p.

Le parc, P. 2017. Introduction à l'informatique. Support de cours : Université de Bretagne occidentale, Faculté des sciences et techniques, Licence 1ére Année. Bretagne, 23p.

Pierrot peladeau. (1988). « Chroniques des sociétés surveillées, lettre d'une des Amériques » Bulletin de liaison du CREIS n°5.

Postel, M (2004). Introduction au logiciel MATLAB. Paris, Université pierre et Marie curie. Laboratoire Jacques-Louis Lions, 23p.

Prescott, M., Harley, P (2002). Microbiology. 5° Ed. The McGraw-Hill Companies.1139p.

Richer, J. (2008). Bioinformatique BTV Alignement de séquences unité de formation et de recherche, 60P. Disponible dans : http://WWW.info.univ-anger.fr/Pub/richer.

Slaouti, A (2002). La revue des sciences commerciales, méthodologie d'identification des systèmes d'informations pertinents, INC, numéro 01, 111p.

Snustad D., Simmons., M .2006 . Transcription and RNA processing. Principles of genetics. Wiley : 279-310 p.

Taconet, C., Conan, D., Bac, C.2015, Conception et programmation. Département INF. Paris[Enligne].(Pageconsulté :16/6/2019).http://www-inf.it-sudparis.eu/COURS/CSC4002/EnLigne/Cours/CoursUML/3.1.html

Thevenont, J. (1985). L'intégration des caractéristiques organisationnelles dans la conception du système d'information. Thèse de doctorat: science de gestion. Université de Montpellier ,45p.

Thompson, J., Higgins, D., Gibson, T. (1994) "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice" Nucleic acids Res. 22, 4673-4680.

Tissseau, J. (2009). Initiation à l'algorithmique. Ecole nationale d'ingénieurs de Brest.

Watson, J., Baker, T., Gann, A. et al. (2012). «Biologie moléculaire du gène 6°éd». France: Pearson. 688p.

Watson J.D., **Crick F.H.C., 1953 .**19"A Structure for Desoxyribose Nucleic Acid. 171: 737-738 p.

Zeigler, B. Kim, D. Praehofer, H. (2000). Theory of Modeling and simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems.

Zeigler, B. (1976). Theory of Modeling and Simulation.

Année universitaire : 2018/2019

Présenté par : Chaabani Aya
Douadi Khaoula

Intitulé : **Modélisation du processus de la traduction d'une séquence d'ADN naturelle et d'une séquence chimère en séquence protéique**

Mémoire de fin de cycle pour l'obtention du diplôme de Master en **Mycologie et Biotechnologie fongique**

Résumé:

Ce travail a été réalisé dans le but de développer un logiciel de modélisation de l'information génétique en protéines. Nous avons pu mettre au point un logiciel qui a la capacité de lire des séquences d'ADN qu'elles soient réelles (qui existent dans la nature) ou chimères (imaginées). Cette modélisation a été implémentée dans le langage MATLAB. Le logiciel a été par la suite vérifié et validé. D'après les résultats, on peut dire que notre logiciel possède la capacité de simuler le processus naturel de la traduction des séquences d'ADN en protéines. Ainsi, le logiciel développé est capable de traduire des séquences d'ADN chimère en protéines et peut ainsi servir dans les recherches de plusieurs domaines tels que le domaine : pharmaceutique, cosmétique et industrie.

Mots clés : ADN ; ARN ; Protéines ; Acide aminés ; Séquence chimère ; Programme ; Modélisation ; Simulation ; Alignement.

Laboratoire de recherche : /

Jury d'évaluation :

Présidente du jury : Mme. Abdelaziz Ouidad (Maitre de conférences B -UFM Constantine)

Rapporteur : Mme. Djama Ouahiba (Maitre assistante A -UFM Constantine)
Dr. Arabet Dallel (Maitre de conférences A-UFM Constantine)

Examinatrice : Dr. Chehili Hamza (Maitre de conférences B- UFM Constantine)

Date de soutenance : 14/07/2019