Le stockage de l'information dans les polymères

Théo ONDOUA, Pierre CHARITAT et Romain CLERBOUT

Année scolaire 2016-2017

Table des matières

Introduction **3**

1 Mise en place et utilisation **4**

1.1 Pourquoi utiliser les polymères ? 5

1.1.1 Leur taille 5

1.2 Pourquoi les polymères classiques ? 5

1.2.1 Durabilité de l'information 6

1.2.2 Utilisation commerciale 6

1.2.3 Inconvénients 7

1.3 La synthèse de l'ADN 7

1.3.1 Le pré-séquençage de l'ADN 8

1.3.2 Le séquençage de l'ADN 9

1.3.3 PCR 9

1.3.4 Le décodage de l'ADN 10

1.3.5 Application pour les polymères synthétiques 11

2 Le stockage de l'information et sa représentation avec 4 bases **12**

2.1 Le stockage de l'information 12

2.1.1 Qu'est-ce que le binaire ? 13

2.1.2 Pourquoi utilise-t-on le binaire ? 13

2.1.3 Comment l'ordinateur lit ou écrit en binaire ? 14
2.2 Stockage d'information sur l'ADN 15
2.2.1 Conversion binaire-ADN 15
2.2.2 Minimisation du risque d'erreur 16
2.2.3 Indexation des segments 17
2.2.4 Inconvénients 18
2.2.5 Algorithmique 19

Conclusion **21**

A Procédure détaillée **23**

B Un exemple **25**

C Tableau d'encodage **29**

Introduction

La quantité d'information à stocker augmente considérablement depuis une dizaine d'années et l'explosion du numérique. Google, Facebook ou YouTube utilisent de gigantesques datacenters, de grandes salles remplies de disques durs, afin de stocker les énormes quantités d'informations que ces plateformes génèrent. On estime que la quantité d'informations stockée dans le monde entier atteindrait 16 zettaoctets (10^{21} octets) cette année [7]. De plus, ces datacenters consomment énormément d'énergie, produisent beaucoup de chaleur et prennent beaucoup de place. Les scientifiques se penchent donc sur de nouvelles méthodes pour stocker les données. L'une d'elle est de stocker l'information à l'échelle moléculaire, sur des polymères. Un polymère est une grande molécule composé d'éléments de base, appelés monomères reliés entre eux par des liaisons covalentes. On trouve des polymères dans la nature, comme l'ADN où les monomères sont des nucléotides, mais les scientifiques arrivent également à créer des polymères synthéthiques, comme le polystyrène composé uniquement de styrène. Cette méthode permettrait de réduire drastiquement la taille nécessaire pour stocker l'information. Dans le cadre de ce TPE, nous avons pris comme exemple l'ADN et avons travaillé sur cet article [2] qui résulte du travail de plusieurs chercheurs. Nous avons également pu rencontrer Dr. Jean-François Lutz et son équipe qui travaillent sur le stockage de l'information sur des polymères synthétiques classiques [1]. **Nous nous demanderons alors si les polymères ne seraient pas le disque dur de demain.** Nous étudierons donc dans un premier temps comment les polymères, avec comme exemple l'ADN, sont synthétisés, modifiés et lus, ainsi que les avantages et les inconvénients de ces derniers. Dans un second temps, nous nous intéresserons au stockage de l'information dans nos ordinateurs, et à la méthode développée par les scientifiques [2] pour transformer cette information en une séquence de nucléotides, ainsi que les avantages et les inconvénients de cette méthode.

Chapitre 1

Mise en place et utilisation

Nous nous pencherons ainsi sur les polymères comme support de l'information numérique. Les polymères sont des macromolécules composées d'une chaîne de molécules semblables appelées monomères. L'ADN ou Acide Désoxyribo Nucléïque qui est le principal composant des chromosomes et qui se trouve à l'intérieur des noyaux des cellules vivantes est un polymère naturel. En effet, il contient toutes les informations nécessaires à la vie d'un organisme, codées avec seulement les bases azotées suivantes : Adénine, Guanine, Thymine et Cytosine appelées nucléotides qui sont les monomères de l'ADN. Ces macromolécules ont en effet un grand potentiel dans ce domaine étant donné leurs divers avantages. Nous parlerons des multiples options que les polymères peuvent apporter par rapport aux méthodes de stockage utilisées actuellement mais aussi des possibles inconvénients qu'ils peuvent poser et donc de leur utilisation possible en général car il faut rappeler que ce processus n'est pas encore commercialisé. Il est certes beaucoup plus attrayant sur plusieurs plans que nous étudierons mais il se pose en parallèle un problème de coût lié à l'utilisation des polymères. En effet la mise en place de cette méthode de stockage est un élément complexe de part sa production qui passe par la synthèse de ces polymères mais également par le temps d'écriture et de lecture dont nous parlerons également dans la suite de notre sujet. C'est donc avec ces données que nous allons parler du stockage de l'information dans les polymères par le biais de la synthèse de ceux-ci mais aussi de l'utilisation de ce stockage, qu'elle soit positive ou négative.

1.1 Pourquoi utiliser les polymères ?

Nous allons tout d'abord nous intéresser à l'utilisation que ce stockage de donnée peut permettre, que ce soit dans l'ADN ou dans les polymères en général. Pour ce faire, nous avons rencontré à l'institut Charles Saudron Dr. Jean-François Lutz, directeur de recherche au CNRS. Tout d'abord, il faut savoir que la demande de stockage de données est de plus en plus importante au fil des années mais que la capacité de stockage numérique ne suit pas. Notre génération actuelle crée d'ailleurs plus de données que toute l'humanité ne l'a jamais fait auparavant.

1.1.1 Leur taille

Ainsi, l'utilisation des polymères pour archiver toutes ces informations est une option intéressante car ces supports sont extrêmement denses avec, dans le cas de l'ADN, une limite d'un exabyte [109 Go/mm^3] (un octet correspond à 8 bits). Nous avons donc un important stockage qui se fait à l'échelle nanométrique, ce qui implique évidemment un gain de place énorme. Pour donner une idée de l'espace nécessaire pour une quantité x de données, nous pouvons nous rappeler les datacenters d'entreprises telles que Google qui demandent énormément de place et de matériel pour des archives. En comparaison, avec l'utilisation de polymères, tout le savoir humain pourrait être retenu dans l'équivalent d'une ou deux armoires. Cette évolution incroyable est permise par la possibilité de stocker des données sur des chaînes de polymères.

1.2 Pourquoi les polymères classiques ?

Pour les polymères classiques tels que le polyester, la technique consiste à traduire notre information (texte, image, musique...) en une séquence binaire. Les scientifiques vont ensuite implanter cette séquence dans le polymère en plaçant les monomères dans un certain ordre, la plupart des polymères étant composé de deux types de monomères différents, ils prennent un monomère pour le chiffre 0 et un monomère pour le chiffre 1. Cette méthode est donc plus efficace car comme nous l'avons précisé précédemment, nous n'avons pas 4 mais 2 monomères, ce qui rend les modifications avec le binaire plus simples. Nous procédons alors à un séquençage, puis une analyse pour retrouver le message (la lecture).

1.2.1 Durabilité de l'information

Cette technique de stockage est aussi un moyen de conserver cette information beaucoup plus longtemps. En effet un disque dur conserve une information entre 5 et 10 ans alors que sur un polymère l'information peut être conservée jusqu'à 1 million d'années car des matériaux comme le polyester se dégradent plus lentement. L'ADN se conserve par exemple pendant plus de 500 ans. La différence entre l'ADN et le polyester est certes grande mais cela constitue tout de même un changement conséquent par rapport aux disques durs. Nous pouvons ainsi envisager la possibilité d'avoir plus d'espace pour les données d'archives et par conséquent libérer beaucoup plus d'espace pour les données actuelles et qui sont destinées à être accessible rapidement. Cette évolution permet également des avancées scientifiques dans des domaines comme la conquête spatiale où les datas humaines pourront être conservées longtemps et donc permettre de transporter plus d'informations plus longtemps. Nous revenons alors à l'idée que cette technologie soit un projet à long terme qui sera efficace dans plusieurs années, en espérant que le domaine du stockage d'informations dans les polymères soit plus avancé et perfectionné.

1.2.2 Utilisation commerciale

Dr. Jean-François Lutz nous a également parlé d'une utilisation plus commerciale de cette technique. En effet, les données étant stockées dans des polymères, nous pourrions envisager l'éventualité que ces polymères soient incorporés directement dans certains produits à la fabrication pour permettre au commerçant de prouver l'appartenance du produit à telle ou telle entreprise. Cela permettrait d'éviter les contrefaçons grâce à ces implantations directement liées à l'entreprise. L'intérêt par rapport à une puce qui est le moyen principal utilisé en ce moment, est l'incorporation direct des polymères dans le produit contrairement à la puce qui prend du coup plus de place. Cela permet aussi d'éviter toute détérioration ou piratage car contrairement à une puce, les polymères ne sont pas électroniques. Cette vision du stockage plus attractive pour les commerçants apporte donc un autre avantage du stockage de données dans les polymères.

1.2.3 Inconvénients

En revanche, ce système de stockage a quelques inconvénients. Tout d'abord, son utilisation est lente par rapport à la vitesse d'écriture et de lecture. L'écriture met en effet entre 1 et 2 heures pour une simple phrase à stocker, et la lecture quelques minutes pour la phrase correspondante. C'est pour cette raison que pour l'instant, les polymères ne sont vus que comme un moyen de stockage d'archive et non de stockage direct. La technologie actuelle ne permettant pas d'aller plus vite, c'est pour cette raison que l'on n'entend pas encore vraiment parler de ce stockage. Cependant nous pouvons espérer une avancée bénéfique car, comme nous l'a précisé le Dr. Jean-François Lutz, il y a seulement 10 ans, l'écriture d'une simple phrase mettait plusieurs semaines.

1.3 La synthèse de l'ADN

Pour synthétiser une séquence nucléotidique, les scientifiques commencent par entrer la séquence d'ADN obtenue par une opération, dont nous parlerons par la suite, dans un logiciel sur un ordinateur connecté à un synthétiseur d'oligonucléotides qui va procéder à l'assemblage des nucléotides in vitro.

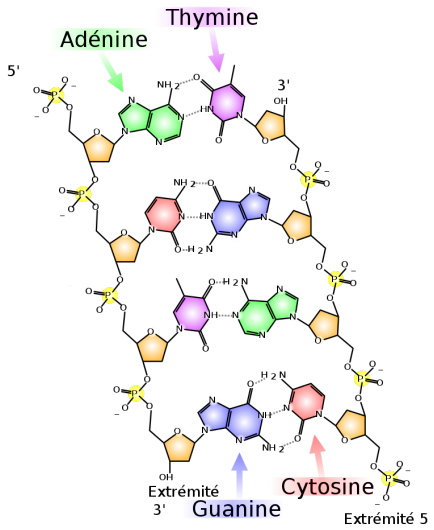

FIGURE 1.1 – Structure des nucléotides

1.3.1 Le pré-séquençage de l'ADN

Ce synthétiseur est connecté à plusieurs bouteilles contenant différentes solutions chimiques dont celles des quatres nucléotides A, T, C et G et celle d'une solution de lavage. Le pré-séquençage (assemblage des nucléotides) qui suit un sens 3' vers 5', à l'inverse de ce que les enzymes pourraient faire dans le séquençage naturel, se passe dans une fiole dont une des extrémités est un support appelé résine (composé de polymères insolubles dans tous les solvant présents lors de la synthèse) où viendra se greffer le premier nucléotide. Suite à la greffe de celui-ci sur la résine grâce à une réaction chimique, un rinçage avec la solution de lavage va retirer l'excédent qui n'aura pas réagi avec la résine. Puis ce sera au tour du deuxième nucléotide qui va se greffer au premier grâce au même procédé et ainsi de suite... Ce procédé s'appelle la synthèse en phase solide et contient donc une réaction chimique à chaque étape. Durant le processus de synthèse, il est possible d'avoir des erreurs dans les séquences, ces erreurs sont d'approximativement une pour 500 bases synthétisées. Elles seront réparées grâce à un processus d'encodage/décodage. Les séquences synthétisées avec ce procédé ont une longueur d'environ 15 à 25 nucléotides.

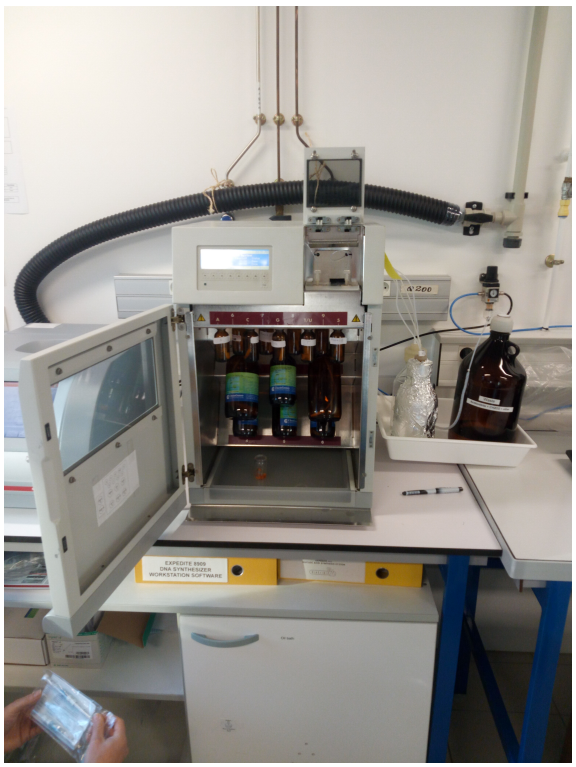

FIGURE 1.2 – Synthétiseur d'oligonucléotides

1.3.2 Le séquençage de l'ADN

Suite au pré-séquençage de tous les nucléotides dont nous avons besoin pour la séquence (que l'on appellera bibliothèque) que nous voulons stocker, les différentes parties de cette séquence sont clivées de leur résine et sont assemblées pour créer une seule séquence contenant le fichier. La séquence reconstituée est amplifiée avec une PCR (expliqué plus tard dans la partie1.3.3) d'un total de 22 cycles en utilisant des conditions de thermocycleur sélectionnées pour un traitement A / T / G / C uniforme44. L'amplification par PCR permet l'enrichissement des nucléotides avec des adaptateurs 5 'et 3' correctement synthétisés et nous a permis d'obtenir une concentration appropriée pour le séquençage [3]. La bibliothèque amplifiée a été purifiée, quantifiée sur le Bioanalyseur Agilent 2100 (Le bioanalyseur Agilent 2100 constitue la méthode standard pour l'analyse et le contrôle de la qualité de fragments d'ADN) puis diluée à une concentration de 16 pM (picomolar : 10-12 mol/m3) pour enfin être séquencée dans une machine appelée Illumia HiSeq 2000. Cette machine permet le séquençage de plusieurs millions de fragments d'ADN, elle sert donc au séquençage à haut débit. La technologie utilisée repose sur une amplification des fragments d'ADN suivi d'un séquençage des colonies d'ADN. Pour simplifier, ce processus identifie les bases d'ADN et les incorpore à une chaîne d'acide nucléique. La réaction de séquençage utilisera seulement 0,1 % de la bibliothèque.

1.3.3 PCR

Lorsque les chercheurs choisissent l'ADN comme polymère de stockage, ils utilisent la PCR [6]. La *Polymerase Chain Reaction* est une méthode de biologie moléculaire. Elle permet, par amplification génétique in vitro, de dupliquer en grand nombre une séquence d'ADN avec des oligonucléotides et quelques picogrammes d'acide nucléique. La PCR se base sur la répétition de cycles de transitions de température. Il faut tout d'abord passer une étape de chauffage de 10 à 15 minutes à 95°C afin d'homogénéiser le milieu réactionnel par agitation thermique et également d'activer les polymérases *Hot Start* (enzymes qui possèdent une molécule qui est enlevée avec un chauffage à 95 degrés). Les scientifiques commencent ensuite réellement le PCR avec la dénaturation, étape d'environ 1 minute à 95°C qui permet d'homogénéiser le milieu réactionnel et de décrocher les polymérases qui seraient encore liées à une matrice (double brin d'ADN que l'on souhaite dupliquer). Ils poursuivent la réaction avec l'hybridation qui a pour but de croiser les amorces sens et anti-sens aux ADN matrice grâce à la température qui leur est favorable (56 à 64°C) avec leurs propriétés d'hybridations des brins complémentaires d'ADN.

La 3e étape est l'élongation, généralement de 4 à 120 secondes à 72°C, qui permet aux polymérases de synthétiser le brin complémentaire de l'ADN matrice. Cette étape dépend de la longueur de l'amplicon (la séquence amplifiée). Il faut ensuite répéter ce processus sur plusieurs cycles pour obtenir plusieurs brin d'ADN synthétiques.

1.3.4 Le décodage de l'ADN

Cette séquence synthétisée par le procédé précédent doit maintenant être décodée. On utilise pour cela un spectromètre de masse [5]. Un spectromètre de masse est composé d'un système d'introduction de l'échantillon, d'une source d'ions appelée chambre d'ionisation, d'un analyseur et enfin d'un détecteur qui détecte les ions sortant de l'analyseur en fonction de leur masse et qui les retransmet sous forme d'un spectre de masse. Chaque étape se passant sous vide. La spectrométrie de masse permet de détecter et d'identifier des molécules auxquelles nous nous intéressons par mesure de leur masse, ainsi que de caractériser leur structure chimique (moins utile dans notre cas). Son principe réside dans la séparation des molécules chargées (ions) en fonction de leur rapport masse/charge (m/z). Figure 1.3.

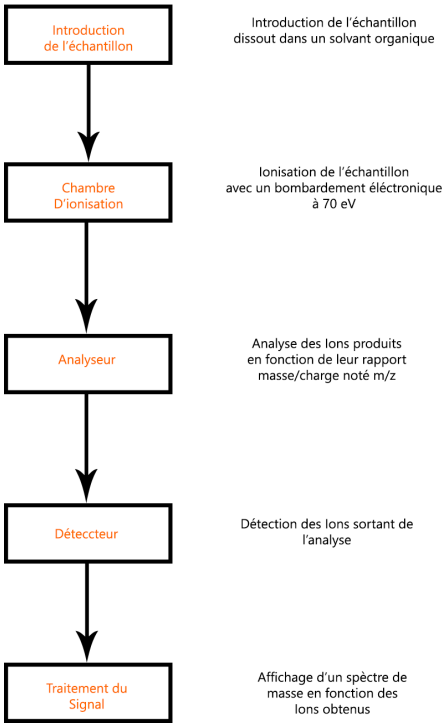

FIGURE 1.3 – Schéma du fonctionnement d'un spéctromètre de masse

La séquence nucléotidique introduite dans le spectromètre de masse est d'abord ionisée par un bombardement électronique a 70eV. L'ion obtenu par ce procédé pour chacun des nucléotides, appelé ion moléculaire, permet de déterminer la masse molaire de chacun de ceux-ci et donc de les reconnaître et de décoder la séquence.

1.3.5 Application pour les polymères synthétiques

Tout ce procédé permet aussi de séquencer les polymères tels que le polyuréthane, les polyamides ou le polyphosphate. La méthode utilisée est la même que pour l'ADN mais seules deux molécules (les monomères de ces polymères) sont utilisées et celles-ci représentent soit un 0 soit un 1 ce qui correspond au code binaire mais sans les erreurs possibles suite aux homopolymères (répétition du même nucléotide). Le procédé est le même avec la première molécule qui se greffe à la résine puis la deuxième à la première etc... Quant au décodage, la spectrométrie de masse sera aussi utilisée.

Chapitre 2

Le stockage de l'information et sa représentation avec 4 bases

Nous avons donc pu voir comment sont créés les brins d'ADN, et que cette synthèse peut provoquer un grand nombre d'erreurs. Nous allons maintenant nous intéresser au stockage de l'information, en général, puis informatiquement. Enfin, nous étudierons plus particulièrement la méthode que les chercheurs ont développé [2], qui permet de transformer une information telle qu'elle est stockée dans un ordinateur en une séquence de nucléotides qui pourra être stockée sur de l'ADN comme nous l'avons vu précedemment. Nous verrons comment cette méthode permet de réduire le risque d'erreur, puis les limites et les inconvénients de cette méthode.

2.1 Le stockage de l'information

Le stockage de l'information est apparu avec les débuts de l'écriture. Les premiers supports de stockage connus furent les tablettes cunéiformes datant de 3400 av. J-C. L'information fut donc d'abord du texte conservée sur du papier (papyrus, vers 3000 av. J-C), puis des sons avec des appareils comme le phonographe (1877). En 1800, les cartes perforées commencent à être utilisées pour programmer des machines (machines à tisser) ou pour le recensement (États-Unis). Avec l'arrivée de l'informatique, de nouveaux supports plus performants sont apparus, comme la bande magnétique, qui repose sur le même principe que la plupart des disques durs d'aujourd'hui. Nous trouvons également les supports optiques tels que les CD ou les DVD, et enfin, les supports flash (clé USB, carte SD, disque dur SSD) qui constituent la dernière génération des technologies de stockage d'information.

Les 3 derniers supports de stockage d'information cités stockent les données en binaire afin qu'un ordinateur puissent facilement les transcrire ou les interpréter. En effet ces supports vont facilement représenter mécaniquement deux états. Pour un courant électrique, nous trouvons soit l'état où le courant passe, soit l'état où le courant ne passe pas par exemple.

2.1.1 Qu'est-ce que le binaire ?

Notre système numérique habituel utilise un système décimal ou base 10, ce qui signifie que nous utilisons 10 chiffres pour représenter les nombres : {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}. Ce sytème s'est imposé du fait des dix doigts de notre main. Il existe d'autres systèmes de numération comme l'hexadécimale ou base 16 qui utilise 16 chiffres : {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f} ou encore le binaire ou base 2 qui utilise 2 chiffres : {0, 1}. Quelques exemples sont donnés dans le tableau 2.1.

Table 2.1 – Conversion décimal/binaire.

Décimal	Binaire
1	1
2	10
3	11
4	100
5	101
6	110
7	111

2.1.2 Pourquoi utilise-t-on le binaire ?

Ce système a été adopté car nous pouvons facilement le réaliser physiquement. Les disque durs, un support magnétique, stockent l'information en créant des couches magnétiques composées de 2 types d'éléments magnétiques mis bout-à bout. Une tête de lecture va passer au dessus de ces éléments qui vont générer un courant positif ou négatif selon le type de l'élément. Pour un signal négatif, l'ordinateur lit 0, et pour un signal positif, l'ordinateur lit 1. Un support optique est composé d'une surface composée de creux. Pour un creux, l'ordinateur lit un 0, sinon il lit un 1.

2.1.3 Comment l'ordinateur lit ou écrit en binaire ?

Un chiffre binaire (0 ou 1) est appelé bits (contracté b), la contraction de binary digits. L'ordinateur lit une information sur un certain nombre de bits. L'information est codée en « mots » qui sont tous de la même taille pour faciliter l'interprétation. Dans une majorité de fichiers, l'information est codée avec des mots de 8 bits (octets en français, contracté o, et bytes en anglais, contracté B). Ces mots permettent une diversité de $2^8 = 256$ possibilités différentes. Cette unité de taille est devenu la norme (la taille des fichier est mesurée en octet en France, en bytes dans le reste du monde). La majorité des ordinateur peuvent coder sur des mots de plus grandes longueurs toujours des multiples de 8 pour calculer plus facilement les tailles de fichiers : 32 ou 64 bits pour les plus récents.

La plupart des fichiers couramment utilisés en informatique sont codés en octets (fichier texte, musique, PDF). Pour l'encodage d'un fichier de texte par exemple, chaque symbole correspond à un nombre entre 0 et 255 représenté en binaire. Il existe différents types d'encodage (ASCII, Unicode...) qui possèdent tous leurs avantages et leurs inconvénients. Un fichier texte d'une page mesure environ 25 kilooctets (1 ko = 1000 octets), une image, environ 1 Mo, un morceau de musique environ 4 Mo (pour une durée de 4 minute) et un film environ 2-3 Go (selon la durée et la qualité de l'image).

Aujourd'hui la plupart des disques durs sont magnétiques et les plus récents sont des mémoires flashs (qui utilisent des semi-conducteurs et sont plus rapides). Cependant ces équipements prennent de la place, et consomment beaucoup d'énergie alors que le nombre d'informations à stocker augmente de plus en plus vite avec l'explosion du numérique (il devrait atteindre 16 zettaoctets = 16.10^{21} octets, en 2017 [7]).

Les chercheurs se penchent donc sur de nouvelles méthodes plus efficaces pour stocker de l'information. Les polymères sont une bonne solution du fait de leur grande capacité de stockage sur des petites surfaces. Ils consommeraient également moins d'énergie, car une machine permettant de les synthétiser ne consomme pas beaucoup.

Stocker des informations sur des molécules pourrait être une solution pour atteindre des tailles plus petites. De plus, seuls les machines servant à écrire/lire/modifier l'ADN consomment de l'énergie.

2.2 Stockage d'information sur l'ADN

Comme nous l'avons vu précedemment, la synthèse de l'ADN peut provoquer des erreurs, et même si elles sont déjà minimisées grâce au PCR notamment, elles doivent tout de même être prises en compte lorsqu'on encode des gros fichiers. De plus, l'ADN se compose uniquement de nucléotides (A, C, G, T) il faut donc trouver une méthode permettant de transformer une séquence binaire en une séquence ADN. Dans l'article que nous avons étudié, les scientifiques ont trouvé une méthode assez complexe mais très efficace. Dans cette partie, nous expliquerons les principes généraux de construction de la séquence ADN qui va coder une information selon la méthode développée et détaillée dans [2]. Un exemple détaillé est donné en annexe A.

2.2.1 Conversion binaire-ADN

On appelle trit, chiffre en base 3 (contraction de ternary digit)

La première idée qui viendrait à l'esprit pour transformer un code binaire en séquence ADN serait de le transformer en base 4 (qui utilise 0, 1, 2 ou 3) puis de remplacer 0 par A, 1 par T, 2 par G et 3 par C. Cependant le système le plus rapide pour synthétiser de l'ADN peut provoquer des erreurs lorsqu'ils doit synthétiser des homopolymères (séquence de monomères identiques, donc pour l'ADN plusieurs fois la même base, par exemple : AAGTTT). Afin d'éviter cela, le séquence binaire est transformée en séquence de base 3 (0, 1 ou 2). Pour encoder chaque trit en un nucléotide, nous regardons le nucléotide précédemment encodé et nous convertissons le trit en une base d'ADN grâce au tableau 2.2.

Table 2.2 – Conversion des trits en base.

pre nt \ trit	0	1	2
A	C	G	T
T	A	C	G
G	T	A	C
C	G	T	A

Par convention, nous décidons que si le trit à encoder est le premier de la séquence, le nucléotide précédent est défini comme étant A ce qui fait qu'une séquence ne pourra jamais commencer par un A. Cette méthode permet donc d'éviter une répétition d'une même base dans la séquence de nucléotide.

2.2.2 Minimisation du risque d'erreur

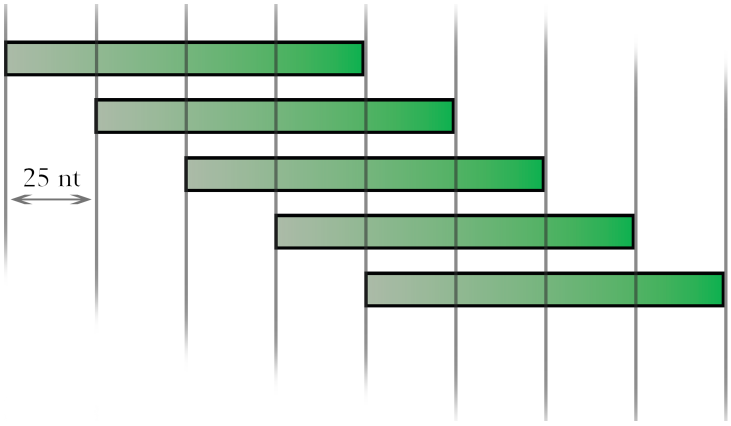

FIGURE 2.1 – Représentation schématique de 5 segments de 100 nt (en vert) où l'on montre la répétition des sous-segments.

Afin de minimiser le risque d'erreur dans le codage d'une séquence binaire S_0, les scientifiques ont imaginé un sytème de répétition complexe. La séquence binaire S_0, convertie en séquence de paires de bases S_5 (voir détails en annexe), va être représentée par une succession de segments de 100 nucléotides, notée F_i, chacuns constitués de 4 sous-segments de 25 nucléotides. Ces segments sont construits de façon à ce que les 3 premiers sous-segments de F_{i+1} soient identiques aux 3 derniers segments de F_i (voir figure 2.1). Chaque segment de 25 nucléotides est répété 4 fois : lors du décodage, ces 4 segments vont être comparés.

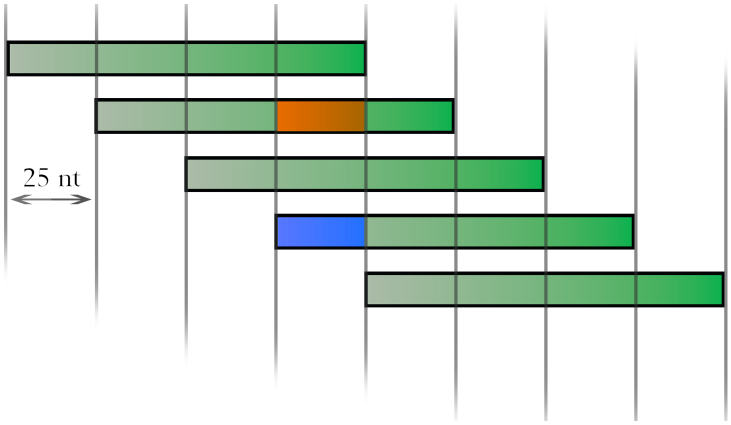

FIGURE 2.2 – Représentation schématique de 5 segments de 100 nt (en vert) dont deux sont endommagés (orange et bleu).

Nous pouvons voir sur la figure 2.2, que deux segments, repérés en bleu et en orange, ont été endommagés. L'ordinateur va comparer les 4 sous-segments et se rendre compte qu'il y en a deux différents et deux exactement identiques. Il va donc décider que ces deux sous-segments identiques correspondent au bon sous-segment. Cependant, il y a encore des risques. Une même erreur peut apparaitre plusieurs fois. Pour éviter cela, un segment de 100 nucléotides sur deux est inversé et complémenté (chaque base est remplacée par son complément). Cette inversion et complémentation est faite directement dans le synthétiseur, pour plus de rapidité, il existe cependant des risques d'erreurs. Ainsi, on rajoute un A ou un T devant le segment, et un C ou un G la fin. Si un segment n'est pas inversé ou n'est pas complémenté alors qu'il devrait l'être, ou est inversé ou complémenté alors qu'il ne devrait pas l'être, ce segment commence donc par C ou G, et nous détectons l'erreur. Cette manipulation permet également d'éviter les homopolymères lorsqu'on met deux segments à la suite, puisque le dernier nucléotide d'un segment ne pourra jamais être le même que le premier.

Cependant afin de créer ces segments qui se superposent, il faut avoir une séquence d'information ternaire dont la longueur est un multiple de 25. Pour cela, nous ajoutons un certain nombre de 0 à cette séquence. Afin de déterminer à quel endroit se termine la séquence contenant l'information et à quel endroit commence la séquence de 0, nous y ajoutons la longueur de la séquence contenant l'information, convertie en ternaire, avec assez de 0 avant pour que sa longueur soit de 20. Nous pouvons ensuite ajouter assez de 0 pour que la séquence entière soit d'une longueur correspondante à un multiple de 25. Ainsi, lors du décodage, l'ordinateur va d'abord lire les 20 dernier trits, puis en déduire la longueur de la séquence originale et va ensuite pouvoir élimininer les 0 en trop.

Ainsi, nous réduisons très fortement le risque d'erreur dans le codage de l'information.

2.2.3 Indexation des segments

Il faut ajouter des informations complémentaires pour chaque segment F_i afin de pouvoir situer ce segment dans la séquence original :

— ID qui indique à quelle séquence binaire S_0 le sous-segment est associé

— Le numéro i du segment F_i (représenté en ternaire par $i3$).

Afin de contrôler les erreurs dans l'indexation, nous utilisons un parity-trit (P).

Très souvent utilisé en informatique (la plupart du temps sous le nom de parity-bit), il consiste à additionner en arithmétique modulaire (voir annexe la somme des chiffres composant l'information à envoyer. Ainsi, lors du décodage, l'ordinateur va également calculer ce parity-trit et vérifier s'il correspond au parity-trit qui a été décodé. Dans le cas présent, seuls les chiffres qui se situent à des positions impaires sont additionnés car chaque trit décodé dépend du nucléotide précédent, nous ne pouvons donc pas éviter une erreur.

2.2.4 Inconvénients

Le procédé que nous venons de voir possède cependant quelques inconvénients, qui limitent le nombre d'informations à stocker, ou laissent quelques possibilités d'erreurs. En effet, la quantité d'information pouvant être convertie en ADN grâce à ce procédé est limitée, pour chaque ID, par la longueur de *i3*, qui ne dépasse pas 12. Pour calculer la longueur de S_1 maximum, nous résolvons cette équation en utilisant la méthode de calcul du nombre de segment donné avec une longueur donnée, avec x_{max} pour longueur maximale :

$$\begin{aligned} & \frac{N_{max}}{25} - 3 = 3^{12} \\ \Rightarrow \quad & \frac{x_{max} + 20}{25} - 3 = 3^{12} \\ & x_{max} = 25(3^{12} + 3) - 20 = 13\ 286\ 080. \end{aligned} \tag{2.1}$$

Lors de la conversion binaire-ternaire (voire procédure détaillée), nous retrouvons environ 1,58 bit/base (ou *trit*). Nous pouvons ainsi stocker environ un maximum de $1,58 * 13\ 286\ 080 = 20\ 992\ 006$ bits. Ce qui nous donne $20\ 992\ 006/8 = 2\ 624\ 000$ o = 2,624 Mo. Nous pouvons multiplier ce résultat par 3^2 puisque nous pouvons répartir des données plus conséquentes en plusieurs séquences S_0, et donc en plusieurs ID : $2,624 * 9 = 23,616$ Mo. Nous nous rendons alors compte que la capacité maximale de ce système est assez limitée (elle correspond à une dixaine de fichier de musique par exemple). Une solution pour augmenter cette capacité maximale serait d'augmenter le nombre de caractères alloués à *i*3 ou ID.

De plus, la minimisation d'erreur n'est pas parfaite puisque les 75 premiers et les 75 derniers nucléotides sont exposés à un risque d'erreur car il y a moins de 4 segments identiques pour vérifier l'intégrité des segments. Nous pouvons également trouver un très faible risque, mais qui doit être pris en considération lors de la conversion de gros fichiers, puisque le système de correction d'erreurs peut être contourné si deux segments possèdent la même erreur, et les deux autres, deux erreurs différentes.

2.2.5 Algorithmique

Nous avons créé, à l'aide du language de programmation Python, deux algorithmes. Le premier permet d'encoder un fichier ou un texte en ADN en suivant toutes les étapes vues précédemment. Le second permet de décoder une séquence ADN encodée précédemment, en un fichier ou un texte. Cet algorithme comporte la correction d'erreurs vu auparavant. Ainsi, nous pouvons tester cette correction en encodant un fichier en ADN, puis en modifiant quelques nucléotides de cette séquence (uniquement des substitutions). Lorsque nous décodons cette séquence, le fichier devrait normalement être correctement décodé. Nous avons implémenté ces algorithmes dans une interface graphique, que vous pouvez télécharger à ce lien, accompagné du code source des deux algorithmes : `https://mega.nz/#!lhhHABjb!PPmXbiOtea4Z68GoF9rNP04pk26XRj-8Wrvt3m9J4xk`. Le programme se présente comme montré dans la figure 2.3 :

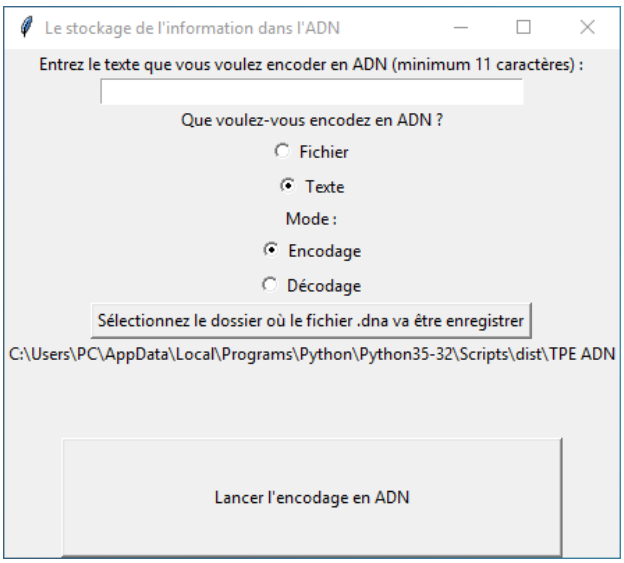

FIGURE 2.3 – Apparence graphique du programme

Cette étape de programmation a permis de mieux comprendre le fonctionnement et l'interêt des étapes de l'encodage.

Conclusion

Dans notre TPE nous avons étudié les polymères et plus précisément le stockage de l'information dans ces derniers. Il y a en effet un marché très ouver car la demande de stockage actuelle est très importante étant donnée l'époque à laquelle nous vivons où la population est à l'origine d'une quantité de données à stocker qui croît très rapidement. Cette demande amène ainsi le domaine scientifique à rechercher de nouvelles méthodes de stockage ce qui nous mène aux polymères. Comme nous avons pu le voir, le stockage dans ces macromolécules permet d'apporter une nouvelle vision de ce dernier. Il apporte de nouvelles possibilités grâce à ses avantages comme la densité de son support ou la durabilité de l'information qui reste un sujet d'actualité avec la conquête spatiale. Par la suite nous nous sommes intéressés à la mise en place de ces supports en prenant comme exemple l'ADN bien que cela soit le même processus en général. C'est d'ailleurs à cause de la synthèse des polymères, même si la science progresse toujours dans ce milieu, que les polymères ne sont pas encore utilisés. Il se pose en effet un problème qu'est le temps. Comme nous avons pu le voir, le temps de séquençage, qui consiste à encoder les données, et celui de lecture sont encore bien trop long par rapport aux formes de stockage que nous utilisons actuellement. Après nous être intéressés à la mise en place et l'utilisation des polymères, nous avons porté notre attention sur l'encodage de l'information. Nous avons alors tout d'abord parlé du stockage numérique de l'information en général à l'aide du binaire puis nous sommes passés au stockage de cette information sur les polymères avec l'exemple de l'ADN, dans le but de rendre notre étude plus simple en nous appuyant sur un exemple concret. Nous avons ensuite vu tout le processus de conversion entre les nucléotides et le binaire puis le stockage final de l'information. Nous pouvons alors conclure en affirmant que le stockage des données dans les polymères possède un énorme potentiel et pourrait bien remplacer les disques durs actuels dans un futur plus ou moins proche.

Cependant, les possibilités au niveau scientifique sont encore insuffisantes pour permettre aux polymères d'être utilisables mais les avancées technologiques laissent penser que cela va de plus en plus évoluer car comme nous l'avait précisé le Dr. Jean-François Lutz, les progrès dans ce domaine sont très rapides.

Annexe A

Procédure détaillée

- Nous récupérons une séquence binaire codé sur 8 bits, avec 256 possibilités, entre 0 et 255.
- Cette séquence est convertie en ternaire de la façon suivante. Pour chaque octet, nous faisons correspondre 5 ou 6 chiffre ternaires (que nous appellerons trit qui est soit 0,1 ou 2) à l'aide du tableau donné en annexe C. Nous obtenons une séquence uniquement composé de 0,1 ou 2 qu'on appellera S_1.
- Nous définissons $len()$ comme étant une fonction qui donne la longueur (en caractère) d'une séquence et on appelle $n = len(S_1)$. Soit S_2 représente n converti en base 3 précéder d'un certain nombre de 0 pour que $len(S_2) = 20$.
- Nous créons ensuite une séquence S_1 qui va contenir assez de 0 pour que $len(S_1) + len(S_2) + len(S_3)$ soit un multiple de 25.
- Enfin nous regroupons ces 3 séquences dans cet ordre : S_1, S_3, S_2. Cela va donner S_4 dont la longueur est un multiple de 25.
- Nous convertissons S_4 en une séquence de nucléotides grâce à la méthode vu précédemment. Cette séquence de nucléotides est nommée S_1.
- Nous définissons $N = len(S_5)$ et ID qui va servir à repérer la séquence binaire d'origine. ID est représenté avec 2 trit, on a donc $3^2 = 9$ possibilités (par exemple 9 Fichiers). Nous allons ensuite diviser S_1 en plusieurs segments de 100 caractères. Ces segments vont se chevaucher de sorte qu'à part les 4 premiers, chaques segments de 25 nucléotides se chevaucheront 4 fois.

Le nombre de segment que nous obtenons est égal à $(N/25) - 3$. Chaque segment va être nommé F_i avec i appartenant à $\{0; (N/25) - 4)\}$.

Chaque segment va être traité séparément :

— Si le i de ce segment est impair, on va inverser la séquence, puis changer n'importe quelle base par son complément (par exemple A devient T, C devient G).
— Nous convertissons i en ternaire ce qui nous donne $i3$, puis on fait précéder $i3$ de assez de 0 pour que $len(i3) = 12$.
— Afin de contrôler que l'indexation ($i3$+ID) est bien synthétisée, on va calculer P, en faisant l'addition des chiffres, placés à un impair, d'ID et de $i3$ en arithmétique modulaire : $ID_1 + i3_1 + i3_3 + i3_5 + i3_7 + i3_9 + i3_{11}$ *modulo* 3. En arithmétique modulaire on ne travaille qu'avec certain nombre, en *modulo* 3 c'est 0,1 ou 2.Nous effectuons l'addition normalement, et si le nombre trouvé n'est pas 0,1 ou 2, on va lui enlever ou ajouter 3, selon s'il est positif ou négatif. Par exemple $5 + 6$ *modulo* 3. On calcule 5 + 6 =11. 11 n'est pas égal à 0, 1 ou 2, on y retranche 3 : 11-3 = 8. On continue cette opération jusqu'à ce que le nombre se trouve entre 0 et 2 : 8-3 = 5 ; 5-3 = 2. On finit donc par trouver 2.
— Nous formons finalement $IX = ID + i3 + P$. Nous transformons IX en ADN grâce à la méthode vue précédemment, en prenant bien en compte le dernier nucléotide du segment et on l'ajoute à ce dernier. On se retrouve alors avec un segment de 115 nucléotides.
— Enfin, nous ajoutons un A un T au début du segment et un C ou un G à la fin. Nous choisissons entre chaque nucléotides aléatoirement si possible, mais en faisant attention à ce que le nucléotides précedent ou suivant ne soit pas le même (erreur lié aux homopolymère).

Annexe B

Un exemple

Nous présentons ici un exemple. Nous avons choisi un texte assez simple de 18 octets : « Théo Romain Pierre ».

1. Nous transformons chaque caractère en binaire, grâce au codage ASCII, puis, à l'aide du tableau donné en annexe C), en trinaire. Les espaces sont représentés par « _ ».

Table B.1 – Conversion caractère/binaire/ternaire

T	h	e	o	_	R	o
01010100	01101000	01100101	01101111	00100000	01010010	01101111
20112	20200	20001	02110	02212	21100	02110

m	a	i	n	_	P	i
01101101	01100001	01101001	01101110	00100000	01010000	01101001
11021	01112	20210	00021	02212	12011	20210

e	r	r	e
01100101	01110010	01110010	01100101
20001	10101	10101	20001

Nous obtenons :

$$\begin{aligned} S_1 &= 201122020020001021100221221100021101102 1 \\ &\quad 0111220210000210221212011202102000110101101012 0001. \end{aligned}$$

2. Cette séquence « mesure » 90 caractères, convertit en base 3 cela nous donne 10100. Nous le faisons précéder par 15 « 0 » :

$$S_2 = 00000000000000010100.$$

La longueur $len\,(S_1 + S_2) = 90 + 20$ est égale à 110. Le plus proche multiple de 25 est donc 125. S_1 est donc composée de $125 - 110 = 15$ fois « 0 » :

$$S_3 = 000000000000000$$

Nous nous retrouvons donc avec :

$$\begin{aligned} S_4 \;=\; & 201122020020001021100221221100021101102101112 \\ & 202100002102212120112021020001101011010120001 \;\; 000000000000000 \\ & 00000000000000010100. \end{aligned}$$

3. Nous convertissons le tout en une séquence de nucléotides :

$$\begin{aligned} S_5 \;=\; & \text{TAGATGTGTATACGACAGACGCAGCAGACGTGAGTCTATCGAGATG} \\ & \text{TGACGTATCGCAGCTGTCTGTGACACGTCTAGTCTAGTCACGTCGT} \\ & \text{ACGTACGTACGTA} \;\; \text{CGTACGTACGTACGTCGACG} \end{aligned}$$

N est égal à 125. Il y aura donc $(N/25) - 3 = 2$ segments allant de 0 à 1. Nous créons ces segments afin qu'ils se chevauchent :

TAGATGTGTATACGACAGACGCAGC AGACGTGAGTCTATCGAGATGTGAC GTATCGCAGCTGTCTGTGACACGTC TAGTCTAGTCACGTCGTACGTACGT
AGACGTGAGTCTATCGAGATGTGAC GTATCGCAGCTGTCTGTGACACGTC TAGTCTAGTCACGTCGTACGTACGT ACGTACGTACGTACGTACGTCGACG

Pour chaque segment, nous inversons et complémentons la séquence en fonction de i :

- Segment 0 :
  $i = 0$, et est donc pair, nous laissons le segment tel qu'il est :

$$\begin{aligned} F_0 \;=\; & \text{AGATGTGTATACGACAGACGCAGCAGACGTGAGTCTATCGAGATGT} \\ & \text{GACGTATCGCAGCTGTCTGTGACACGTCTAGTCTAGTCACGTCGTA} \\ & \text{CGTACGT} \end{aligned}$$

- Segment 1 :
  $i = 1$, et est impair, nous inversons et complémentons donc le segment :

$$F_1 = \text{CGTCGACGTACGTACGTACGTACGTACGTACGTACGTACGACGTGACTAG ACTAGACGTGTCACAGACAGCTGCGATACGTCACATCTCGATAGAC TCACGTCT}$$

Pour chaque segment nous allons définir *i3* :

- Segment 0 :
  Nous convertissons $i = 0$ en base 3 :0 puis nous le faisons précéder d'assez de 0 pour que sa longueur soit 12 : 000000000000.
- Segment 1 :
  Nous convertissons $i = 1$ en base 3 :1 puis nous le faisons précéder d'assez de 0 pour que sa longueur soit 12 : 000000000001

Nous définissons ID aléatoirement parmi les 9 possibilités, pour l'exemple nous allons prendre 21.

Nous calculons P pour chaque segment :

- Segment 0 :
  $ID : 21, i3 : 000000000000$
  $P = ID1 + i3_1 + i3_3 + i3_5 + i3_7 + i3_9 + i3_{11}\ modulo\ 3 = 2 + 0 + 0 + 0 + 0 + 0 + 0\ modulo\ 3 = 2.$
- Segment 1 : $ID : 21, i3 : 000000000001$
  $P = ID1 + i3_1 + i3_3 + i3_5 + i3_7 + i3_9 + i3_{11}\ modulo\ 3 = 2 + 0 + 0 + 0 + 0 + 0 + 0\ modulo\ 3 = 2.$

Nous formons donc IX :

- Segment 0 : $IX = ID + i3 + P$: 210000000000002.
  Nous transformons IX en ADN en prenant en compte le dernier nucléotide du segment, ce qui nous donne : GACGTACGTACGTAT de longueur 15.
- Segment 1 :
  $IX = ID + i3 + P$: 210000000000012
  Nous transformons IX en ADN en prenant en compte le dernier nucléotide du segment, ce qui nous donne : GACGTACGTACGTCA de longueur 15.

Nous ajoutons IX au segment de base :

- Segment 0 :

$$F_0 = \text{TAGATGTGTATACGACAGACGCAGCAGACGTGAGTCTATCGAGAT}$$
$$\text{GTGACGTATCGCAGCTGTCTGTGACACGTCTAGTCTAGTCACGTC}$$
$$\text{GTACGTACGT GACGTACGTACGTAT}$$

- Segment 1 :

$$F_1 = \text{CGTCGACGTACGTACGTACGTACGTACGTACGTACGTACGACGTGACTA}$$
$$\text{GACTAGACGTGTCACAGACAGCTGCGATACGTCACATCTCGATAG}$$
$$\text{ACTCACGTCT GACGTACGTACGTCA}$$

Nous ajoutons enfin deux nucléotides aux extrémités du segment, ce qui forme un segment de 117 nt :

- Segment 0 :

$$F_0 = \text{A TAGATGTGTATACGACAGACGCAGCAGACGTGAGTCTATCGA}$$
$$\text{GATGTGACGTATCGCAGCTGTCTGTGACACGTCTAGTCTAGTCAC}$$
$$\text{GTCGTACGTACGTGACGTACGTACGTAT G}$$

- Segment 1 :

$$F_1 = \text{T CGTCGACGTACGTACGTACGTACGTACGTACGTACGACGTGA}$$
$$\text{CTAGACTAGACGTGTCACAGACAGCTGCGATACGTCACATCTCGA}$$
$$\text{TAGACTCACGTCTGACGTACGTACGTCA C}$$

Annexe C

Tableau d'encodage

Voici un le tableau, donné dans cet article [2], permettant de transformer une séquence binaire, codé sur 8 bits, en une séquence ternaire, codée sur 5 ou 6 trits. Le tableau entier possède 256 lignes, pour les 256 possibilités d'un mot de 8 bits. On y retrouve :

— Le caractère ASCII correspondant à la valeur binaire
— La valeur binaire représenté en décimal
— Le mot de 5 ou 6 trits correspondant à cette valeur binaire
— La longueur du mot ternaire correspondant à la valeur binaire

	Caractère ASCII	Codage binaire	Codage ternaire	Longueur du codage
0	#	0	22201	5
1	U	85	22200	5
2	ª	170	22122	5
3	☐	127	22121	5
4	ý	253	22120	5
5	4	52	22112	5
6		138	22111	5
7	)	41	22110	5
8	V	86	22102	5
9	*	42	22101	5
10	d	100	22100	5
11	,	44	22022	5
12	ú	250	22020	5
13		132	22021	5
14	¡	161	22012	5
15	b	98	22010	5
16	#	8	22002	5
17	"	34	22011	5
18	[NL]	10	22001	5
19		149	22000	5
20	W	87	21222	5
21	#	21	21221	5
22	J	74	21220	5
23	$	36	21212	5
24	E	69	21210	5
25	±	177	21202	5
26	#	20	21211	5
27	Õ	213	21200	5
28	£	163	21201	5
29	å	229	21121	5
30	ÿ	255	21122	5
31	Å	197	21120	5
32		133	21112	5
33	ü	252	21110	5

34	#	26	21111	5
35		173	21101	5
36		151	21102	5
37	R	82	21100	5
38	K	75	21022	5
39	%	37	21021	5
40	¦	166	21011	5
41	¿	191	21020	5
42	X	88	21012	5
43	?	63	21010	5
44	D	68	21001	5
45		150	21002	5
46	L	76	21000	5
47	#	4	20222	5
48		154	20221	5
49	ê	234	20212	5
50	#	22	20220	5
51	¢	162	20211	5
52	i	105	20210	5
53	f	102	20202	5
54	«	171	20201	5
55	h	104	20200	5
56	©	169	20122	5
57	Ä	196	20121	5
58	Đ	208	20120	5
59	T	84	20112	5
60		130	20111	5
61		146	20102	5
62	H	72	20110	5
63	#	16	20101	5
64	B	66	20100	5
65	#	24	20022	5
66	j	106	20012	5
67	ß	223	20020	5
68	:	58	20021	5
69		137	20011	5
70	I	73	20010	5

71	e	101	20001	5
72	¨	168	20002	5
73	µ	181	12221	5
74	¯	175	12222	5
75	û	251	20000	5
76	(	40	12220	5
77		140	12212	5
78	#	17	12211	5
79	S	83	12210	5
80	þ	254	12202	5
81	ð	240	12201	5
82	Ö	214	12200	5
83	5	53	12122	5
84	Ê	202	12112	5
85	#	25	12121	5
86	#	18	12120	5
87	÷	247	12111	5
88	®	174	12110	5
89	p	112	12102	5
90	Y	89	12101	5
91	Ò	210	12100	5
92	Ù	217	12012	5
93	ø	248	12020	5
94	Â	194	12021	5
95	¶	182	12022	5
96	P	80	12011	5
97	O	79	12002	5
98	Ã	195	12010	5
99		49	12001	5
100	Ñ	209	12000	5
101	¥	165	11222	5
102	õ	245	11221	5
103	#	2	11220	5
104	Q	81	11212	5
105	&	38	11211	5
106		141	11202	5
107	Ó	211	11210	5

108	ï	239	11200	5
109	_	95	11201	5
110	+	43	11122	5
111	à	224	11121	5
112	Ë	203	11112	5
113		145	11120	5
114		147	11110	5
115	#	19	11111	5
116	2	50	11101	5
117		136	11102	5
118	k	107	11100	5
119		134	11022	5
120	m	109	11021	5
121		153	11020	5
122		148	11002	5
123	Í	205	11010	5
124	Ô	212	11011	5
125	6	54	11012	5
126	ñ	241	11000	5
127		156	11001	5
128	s	115	10222	5
129	t	116	10221	5
130	N	78	10220	5
131	C	67	10211	5
132	F	70	10212	5
133	²	178	10210	5
134		159	10202	5
135		142	10201	5
136	\	92	10200	5
137	0	48	10122	5
138	Z	90	10120	5
139	Ú	218	10121	5
140	~	126	10112	5
141	'	39	10111	5
142	Û	219	10102	5
143	§	167	10110	5
144	r	114	10101	5

145	¬	172	10022	5
146	#	14	10100	5
147	x	120	10020	5
148		139	10021	5
149		160	10012	5
150	!	33	10011	5
151	³	179	10010	5
152	u	117	10002	5
153	á	225	10001	5
154		129	10000	5
155	·	183	02222	5
156	æ	230	02220	5
157	#	35	02221	5
158	]	93	02210	5
159	#	6	02211	5
160		48	02212	5
161	8	56	02201	5
162		158	02202	5
163	¹	185	02121	5
164	/	47	02122	5
165		143	02200	5
166	{	123	02111	5
167	Ì	204	02120	5
168	ò	242	02112	5
169	o	111	02110	5
170	g	103	02102	5
171	l	108	02101	5
172	[TAB]	9	02100	5
173	A	65	02022	5
174	ù	249	02020	5
175				
	13	02021		5
176	´	180	02012	5
177	â	226	02001	5
178		144	02002	5
179	#	15	02010	5
180	9	57	02011	5

181		128	02000	5
182		135	01220	5
183	ó	243	01221	5
184	¾	190	01222	5
185	Ï	207	01212	5
186	M	77	01211	5
187	-	45	01210	5
188	[	91	01202	5
189	À	192	01201	5
190	º	186	01122	5
191	Ø	216	01200	5
192	a	97	01112	5
193	v	118	01120	5
194	ö	246	01121	5
195	×	215	01111	5
196	3	51	01102	5
197	Î	206	01110	5
198	¸	184	01100	5
199	ã	227	01101	5
200	é	233	01022	5
201	í	237	01021	5
202	¼	188	01020	5
203	q	113	01012	5
204	1	49	01011	5
205	É	201	01010	5
206		155	01002	5
207	Þ	222	01000	5
208	ç	231	01001	5
209	#	5	00222	5
210	#	27	00221	5
211		131	00212	5
212	¤	164	00220	5
213	#	3	00211	5
214	.	46	00210	5
215	w	119	00201	5
216	#	28	00202	5
217	°	176	00200	5

218	#	23	00122	5
219	@	64	00121	5
220		157	00120	5
221	»	187	00112	5
222	ô	244	00110	5
223	î	238	00111	5
224	`	96	00102	5
225	ë	235	00101	5
226	<	60	00022	5
227	#	1	00100	5
228	n	110	00021	5
229	È	200	00011	5
230	Ý	221	00020	5
231	c	99	00012	5
232	#	31	00010	5
233	Æ	198	00002	5
234	Á	193	00001	5
235	}	125	00000	5
236	|	124	222222	6
237		152	222221	6
238	z	122	222220	6
239	G	71	222212	6
240	^	94	222211	6
241	Ü	220	222210	6
242	#	29	222202	6
243	Ç	199	222201	6
244	=	61	222200	6
245	#	11	222122	6
246	ä	228	222121	6
247	>	62	222120	6
248	7	55	222112	6
249	y	121	222111	6
250	#	7	222110	6
251	#	30	222102	6
252	è	232	222101	6
253	½	189	222100	6
254	;	59	222021	6

255	ì	236	222022	6
256	[EXTRA]	n/a	222020	6

Bibliographie

Bibliographie

[1] LUTZ, Jean-François ; OUCHI, Makoto ; LIU, David ; SAWAMOTO Mitsuo. Sequence-controlled Polymers. *Science*, 09 Aug 2013, 1238149.

[2] GOLDMAN, Nick ; BERTONE, Paul Bertone ; CHEN, Siyuan ; DESSIMOZ, Christophe ; LEPROUST, Emily ; SIPOS, Botond ; BIRNEY, Ewan. Towards practical, high-capacity, low-maintenance information storage in synthesized dna. *Nature*, Février 2013, 494(7435), p77–80.

Sitographie

[3] WALES, Jimmy ; SANGER, Larry. *Wikipédia [en ligne]*. Wikipédia, modifié le 18 janvier 2017 [consulté le 1er novembre 2016]. Acide désoxyribonucléique. Disponible à l'adresse : `https://fr.wikipedia.org/wiki/Acide_d%C3%A9soxyribonucl%C3%A9ique`

[4] CHEN, Steve ; HURLEY, Chad ; KARIM, Jawed. *YouTube [en ligne]*. e-penser par ailleurs, 18 décembre 2016 [consulté le 18 décembre 2016]. "Degré 0 de l'informatique - 01 - e-penser numérique ". Disponible à l'adress'e :`https://youtu.be/wK2E0ksbf9I`

[5] Webedia. *Overblog [en ligne]*. Mr Magniez, 5 Octobre 2010 [consulté le 5 février 2017]. "La spéctrométrie de masse". Disponible à l'adresse : `http://www.technobio.fr/article-la-spectrometrie-de-masse-51092779.html`

[6] WALES, Jimmy ; SANGER, Larry. *Wikipédia [en ligne]*. Wikipédia, modifié le 17 février 2017 [consulté le 29 janvier 2017]. Réaction en chaîne par polymérase. Disponible à l'adresse : `https://fr.wikipedia.org/wiki/R%C3%A9action_en_cha%C3%AEne_par_polym%C3%A9rase`

[7] IDC. *IDC : The premier global market intelligence firm [en ligne]*. IDC [consulté le 19 janvier 2017]. "Where in the world is storage infographic". Disponible à

l'adresse :http://www.idc.com/downloads/where_is_storage_infographic_243338.pdf