

Rapport du projet **d'algorithmique :**

Léon KOENIG



Introduction à l'algorithmique

Jean-Marie WURTZ

Table des matières :

Introduction

Traduction de l'ADN

Interface graphique de la traduction

Tableau des codons

Organisation de l'arborescence

Conclusion

L'objectif de ce projet a été de créer un programme en langage python avec une interface graphique à partir des connaissances et de l'expérience que l'on a acquis au cours des cours et TP du semestre dans l'UE *Introduction à l'Algorithmique*. Ce programme a pour but de générer une séquence d'ADN de taille choisie, la convertir en sa séquence complémentaire ou reverse pour enfin la traduire en protéine en fonction des paramètres choisis comme le cadre de lecture à utiliser et si on souhaite avoir une représentation des acides aminés en code à une lettre ou en code à trois lettres. De plus le programme doit pouvoir ouvrir une séquence d'ADN au format FASTA présente sur son ordinateur ainsi que présenter un tableau des codons disponible via le menu.

Tout d'abord j'ai utilisé le code source mis à notre disposition sur moodle. Je l'ai lu attentivement entièrement et j'ai annoté certaines lignes afin de bien le comprendre pour pouvoir l'utiliser correctement et y incruster les éléments manquants. Concentrons-nous par exemple sur l'ajout de la traduction de la séquence d'ADN qui a été la partie la plus difficile pour moi. Il m'a fallu commencer par créer un programme (« dnaTranslator ») contenant une fonction qui me permette d'à partir d'une séquence en attribut puisse la traduire en fonction de 2 autres attributs (cadre de lecture et code à X lettre).

La tâche n'a pas été compliquée, à l'aide d'un dictionnaire pour associer les codons à leur acide aminé et de boucles for et if. Ce qui m'a posé le plus de problème a alors été de déterminer quand la traduction doit s'arrêter pour ne pas tomber sur moins de trois nucléotides car ne pouvant pas traduire cela. Pour cela j'ai réduit la range de deux nucléotides quand le cadre de lecture commence au second nucléotide et de un nucléotide quand il commence au troisième. C'est en essayant plusieurs variantes que je suis enfin tombé sur cette solution.

Il m'a alors fallu créer un programme contenant une classe pour créer l'interface graphique du traducteur. Je me suis inspiré de la méthode utilisée dans « gui_dnaConRev » pour la classe et j'ai ensuite ajouté une fonction « translatedna » qui est reliée aux radioboutons pour traduire la séquence d'ADN à chaque fois que l'on modifie un paramètre et que la séquence soit affichée dans une case de texte. En effet un des aspect que j'ai trouvé le plus compliqué a été la création de classes pour former les widgets qui composent l'interface

graphique du programme. J'ai dû passer beaucoup de temps à me renseigner et bien comprendre comment une classe fonctionne pour ensuite pouvoir en utiliser dans mon programme. Le principe de classe est désormais clair pour moi mais l'utilisation de ces dernières pour générer des widgets me reste un concept flou et difficile.

Pour l'incrustation de l'image d'un tableau de codon j'ai dû modifier le programme « gui_dnamenu » afin d'y ajouter une fonction activable lors de l'activation d'un bouton donc sous forme de commande. Cette fonction crée une nouvelle fenêtre tkinter dans laquelle le tableau des codons est affiché. L'incrustation d'image m'a posé beaucoup de problèmes et après avoir essayé un bon nombre de méthodes, j'ai fini par utiliser la librairie PIL (Python Imaging Library) avec les fonctions « ImageTk » et « Image » qui m'ont permis d'ouvrir l'image et de l'afficher dans la fenêtre. J'ai également eu beaucoup de problèmes avec le chemin de l'image (de même pour l'icone en haut à gauche de l'interface graphique), la solution a été d'utiliser un chemin relatif au lieu d'un chemin absolu car ce dernier aurait fonctionné correctement uniquement sur mon ordinateur mais dès que quelqu'un d'autre lancerait mon programme il y aurait un problème. Le chemin relatif est donc la solution la plus polyvalente.

Au cours de la confection de l'ensemble de mes programmes, le programme principal « gui_appli » était contenu dans le même dossier que les autres programmes or pour le rendu il fallait qu'il soit dans le dossier parent et que les autres soient dans un dossier « src ». Mais je ne pouvais pas juste me contenter de le déplacer dans le dossier parent car en lançant le programme je me suis vite rendu compte que cela ne fonctionnerait pas. Plusieurs méthodes existent pour régler cela et celle que j'ai choisi est de créer un package avec l'ensemble des programmes annexes qui deviennent alors des modules que je dois ensuite importer dans le programme principal au début.

En essayant simplement d'importer les modules cela ne fonctionne pas, il faut préciser où se situent les modules du package en précisant leur chemin relatif (et pas absolu pour les raisons vues jusqu'à présent). Pour cela j'ai utilisé le module « sys » et sa fonction « path » pour ajouter le chemin des modules à celui où python les cherche. Ensuite il suffit d'importer l'ensemble

des fonctions de tous les modules du dossier « src » pour pouvoir les utiliser dans le programme principal. Mon environnement de développement (Virtual Studio Code) ne reconnaissait plus par la suite mes fonctions mais cela fonctionnait tout de même en lançant le programme en dehors, je ne sais pas d'où vient cela mais ce serait intéressant de savoir.

Pour conclure, la confection de ce programme a été très enrichissante, la pratique concrète est très différente de la théorie, je me suis senti beaucoup plus à l'aise et je pense avoir mieux assimilé de cette manière. La documentation à partir du cours a été très utile et des recherches internet m'ont permis de régler beaucoup de soucis que je rencontrais, la documentation et la communauté autour du langage python sur internet est impressionnante et très pratique.

L'apprentissage de l'utilisation de linux a été pour moi une super expérience, la liberté et les possibilités offertes par ce système d'exploitation ainsi que la vision de l'open source associée me plais beaucoup et je suis content d'être désormais capable de l'utiliser.

J'ai pour objectif de faire mes propres projets avec python afin de ne pas oublier tout ce que j'ai appris et de pouvoir même m'améliorer. J'espère avoir l'occasion d'utiliser ce langage plus tard au cours de ma carrière car j'ai beaucoup apprécié travailler avec et surtout découvrir un tout nouveau monde très différent de celui de la biologie ce qui est agréable quand on ne fait que de la biologie pendant trois ans. Ce cours m'a permis d'ouvrir mes horizons et ainsi de m'intéresser à plus de sujets, l'informatique est de nos jours plus que jamais un domaine d'avenir et il est important pour moi d'y mettre un pied même en tant que biologiste.

Annexes :

Les nucléotides : fr.wikipedia.org/wiki/Nucléotide

Le code génétique : fr.wikipedia.org/wiki/Code_génétique

Les codons : fr.wikipedia.org/wiki/Codon Format Fasta :
[fr.wikipedia.org/wiki/FASTA_\(format_de_fichier\)](http://fr.wikipedia.org/wiki/FASTA_(format_de_fichier))

Trouver des séquences génomiques :

- www.ncbi.nlm.nih.gov/genome/guide/human
- hgdownload.cse.ucsc.edu/goldenPath/hg19/chromosomes/