**COMPTE RENDU Python**

1. **Explication du programme**
2. Organisation

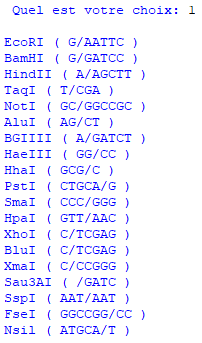
Tout d’abord, nous avons effectué un travail de réflexion sur papier en essayant de schématiser les grandes lignes de notre futur code, nos besoins sous forme de variables et de fonctions. Cette première étape était donc celle de conceptualisation, à l’aide de schémas pour notre première approche.

Nous avons ensuite repris les fonctions conceptualisées et ajouté de nouvelles fonctions en fonction des besoins au fur et à mesure de notre compréhension et de l’avancement du projet. Nous avons aussi mis en commun des idées entre temps, et revu avec plus en précision notre premier essai.

Par la suite, nous avons élaboré la fonction principale, celle de digestion de notre séquence par une enzyme choisie.

Puis ce fût la création de la plupart de nos fonctions, notamment associées au dictionnaire (ajout d’enzyme, vérification enzyme, ouverture fichier, etc…) et à son utilisation (travail de documentation sur les fichiers [1]). Nous avons mis en place le menu et avons aussi travaillé sur la liaison du dictionnaire à la fonction de digestion et l’adapter.

Et enfin, nous avons travaillé sur les détails finaux de notre programme : mise en forme de l’affichage du dictionnaire, vérification de la séquence (séquence ADN valide, ARN, ou invalide), et modification associées notamment dans les choix du menu. Nous avons également modifié la fonction digestion\_multiple pour pouvoir indiquer à l’utilisateur toutes les enzymes mal-écrites lors de l’appel de la fonction.



1. Programme actuel

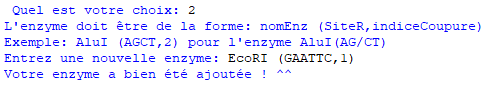
Le choix 1 contient une fonction :

* Afficher\_Dico\_Enz() : fonction qui affiche le dictionnaire

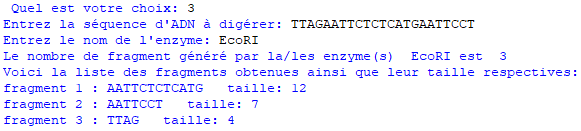
Elle permet la création du dictionnaire, l’ouverture du fichier texte en mode lecture (read), le parcours du fichier ligne par ligne et l’affichage ligne par ligne des enzymes (choix esthétique).

Le choix 2 contient également une fonction :

* Ajout\_Enz() : permet d’ajouter un enzyme au dictionnaire (dans le fichier texte) sous la forme souhaitée.

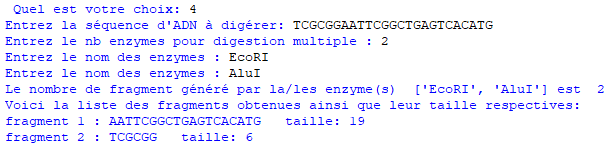


Le choix 3 contient 7 fonctions :

* creation\_dico() : permet de créer le dictionnaire.
* verif\_Enz(enzyme, dico) : Vérifie que l’enzyme appartient au dictionnaire.
* verif\_seq(seq) : Vérifie la validité de la séquence entrée, cela doit-être une séquence d’ADN.
* trouver\_siteR(w, seq\_siteR) : Permet de rechercher si le site de restriction de l’enzyme choisie est présent dans la séquence d’ADN donnée préalablement.
* trie\_liste\_frag(liste) : Trie les différents fragments de la liste précédente en fonction de leur taille. Ils seront triés comme lors d’une migration, c’est-à-dire du plus grand au plus petit.
* afficheResultDig(enz, liste) : Permet d’afficher à l’utilisateur le nombre de fragments obtenus ainsi que la liste triées des fragments.
* digestion(ENZ, seq, dico) : Permet de digérer une séquence d’ADN par une enzyme choisie et de garder dans une liste les fragments.

Le choix 4 contient l’intégralité des fonctions du choix 3 plus une autre :

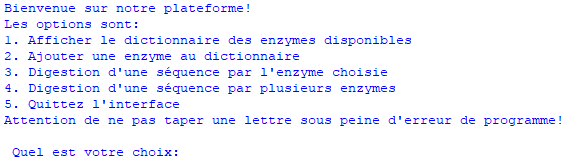
* digestion\_multiple (l\_enz,seq,dico) : Permet de gérer le cas où l’utilisateur effectue plusieurs digestions à la suite.



Le choix 5 permet de quitter le programme.



Notre fonction principale est la fonction Menu(). Elle permet de gérer les choix de l’utilisateur et le bon fonctionnement de l’intégralité de notre programme.



1. **Problèmes rencontrés et solutions associées (améliorations)**
2. Pour l’utilisation du dictionnaire nous avons hésité entre e choix d’un fichier ou de le faire dans le programme directement. Le problème qui se soulevait dans le deuxième cas était que le dictionnaire ne pouvait être modifié, donc cela ne permettait pas l’ajout ni la suppression d’enzymes.
3. Lors de la digestion multiple, les différentes enzymes digèrent la séquence l’une après l’autre dans l’ordre de sélection et pas dans toutes en même temps comme le modèle physique réel.
4. Problème de sauvegarde, perte des fonctions suivantes: creation\_dico(), Ajout\_Enz(), Afficher\_Dico\_Enz(), verif\_Enz(), Menu(). Nous avons donc nous remettre à la tâche. **Ctrl+S est maintenant notre meilleur ami 4ever <3 !!!**
5. Nous avons décidé de sortir le dictionnaire de la fonction digestion() pour éviter de le recréer à chaque digestion et nous l’avons donc placé en paramètre de digestion(). Nous avons sorti le test de vérification des enzymes (appartient dico) pour ne pas avoir à la refaire dans la digestion. Nous avons mis en place l’affichage de afficheResultDig() à la fin de toutes les digestion lors de la digestion multiple (et pas à chaque digestion intermédiaire) et supprimé le compteur nb\_frag, car on peut l'obtenir avec la fonction len() appliqué sur la liste de fragment. Nous avons aussi enlevé la fonction trie\_frag de digestion et l’avons mise dans la fonction résultat (obtention des fragments une seule fois). Enfin nous avons fait le choix d’afficher le nom de la dernière enzyme avec une erreur au lieu de toutes celles qui pourraient être fausses.
6. Nous nous sommes ensuite raviser, et avons choisi d’indiquer toutes les enzymes dont le nom aurait été mal écrit afin que l’utilisateur n’ait pas à ré-appeler la fonction autant de fois qu’il doit de faire de modifications en fonction de ses erreurs.
7. **Questionnements supplémentaires**

Nous avions des pistes d’améliorations de notre programme ainsi que de possibles complexifications :

* Faisons nous en sortes que notre programme soit en capacité de gérer les enzymes avec des sites de restriction de type ATNC (N=toutes les bases) ou ATpuG/ATpyG ... ?
* Ajoutons-nous l'option Supprimer\_Enz\_Dico qui permettrait de supprimer une enzyme du dictionnaire?
* Faut-il vérifier la validité du site de restriction lors de l’ajout d’une nouvelle enzyme ?
* Devons-nous prendre en compte les cas où le site de restriction de deux enzymes se chevauche (digestion simultanée) ?