


Voir journal de bord complet:

 Journal de bord projet adn

Le projet qu'on a choisi de réaliser est pour la discipline de SVT des lycéens de Monsieur Guérin. Il se concentre sur l'ADN, plus précisément sur la phylogénétique. L'objectif principal de ce projet était de pouvoir créer un site interactif, et simple à utiliser (contrairement à beaucoup d'autres sites de bioinformatique existants), où les élèves pourraient réaliser des comparaisons entre différentes séquences d'ADN. Pour cela, il fallait qu'on écrive un code qui permet de charger des séquences d'ADN ou de protéines, avant de calculer les indices de proximité entre ces séquences. Le résultat de ce traitement est un arbre de parenté, également connu sous le nom d'arbre phylogénétique, qui représente graphiquement les relations de parenté entre les différentes séquences pour traduire les filiations évolutives, et les montrer visuellement.

Pour l'organisation du travail, on a parlé de ce qu'il fallait faire au début du projet pour qu'on soit tous d'accord sur la répartition des tâches initiales. De plus, Alek et Nicholas sont allés voir Monsieur Guérin à deux reprises pour parler du projet avec lui, et lui poser quelques questions afin de clarifier certains concepts. Isabella, qui a moins d'expérience en programmation qu'Alek et Nicholas, s'est chargée du CSS, ainsi que d'aider avec les recherches et le planning. Quand il le fallait, elle récrivait à certains endroits le code JavaScript ou le code HTML s'il y avait un problème avec la synchronisation, ou un bug, mais c'est majoritairement Alek et Nicholas qui se sont chargés du code. Ils ont écrit les fichiers HTML, JavaScript, et Python, qui comportent notamment le calcul de l'indice de proximité, l'entrée des séquences, et la génération de l'arbre phylogénétique.

Tout d'abord, nous avons commencé par créer une application flask, et entamé un code python pour charger une séquence d'ADN ou de protéine, ainsi qu'une fonction python pour calculer l'indice de proximité de manière séquencée entre deux séquences. Ensuite, nous avons contacté Monsieur Guérin pour lui poser nos questions et avoir plus de détails au sujet du site, pour savoir ce qu'il voulait qu'on y ajoute exactement. En attendant la date de la première rencontre avec le Maître d'ouvrage, nous avons créé une interface basique HTML, ainsi que l'indice de proximité. Quand Alek est allé parler avec Monsieur Guérin, on a appris qu'il fallait incorporer la délétion dans la fonction de calcul d'indice de proximité, ce qui veut dire qu'un décalage de séquences à chaque nouvelle position doit être envisagé (si nous continuions à parcourir les séquences de manière séquentielle). En outre, Nicholas ajouta l'option de charger une séquence d'ADN sur le site web, et créa un fichier JavaScript pour y ajouter des séquences en plus à comparer et générer des séquences de taille aléatoire, pendant qu'Alek et Isabella recherchaient une façon pour améliorer le calcul de l'indice de proximité pour qu'il prenne en compte les mutations par délétion. Simultanément, Isabella commença à travailler sur le design du site web, et s'est mise d'accord avec son groupe pour faire une animation d'une double hélice. Donc pendant qu'Alek travaillait sur l'implémentation de l'algorithme Needleman-Wunsch en python et de la librairie biopython pour pouvoir aligner plusieurs séquences, Isabella commença les recherches sur l'animation en général, et comment faire l'animation de la double hélice. Avant les vacances, Alek et Nicholas sont allés à l'OH de Monsieur Guérin, où ils ont eu une explication de l'alignement de séquences pour pouvoir créer l'arbre phylogénétique, et pendant les vacances, Isabella avait déjà un brouillon CSS pour l'animation et le design du site, qui marchait bien avec le reste des fichiers, et Nicholas créa un nouveau fichier HTML pour recevoir le calcul de l'indice, ainsi que l'arbre quand il serait complété. Au retour des vacances,

Isabella a fait des recherches sur comment faire un arbre phylogénétique, en regardant les différentes méthodes, ainsi que la méthode neighbor joining en plus de détails, pendant qu'Alek et Nicholas cherchèrent comment le faire en code et comment l'implémenter avec de la visualisation 3D de data ou bien librairie biopython. En outre, Alek implémenta les calculs de proximité par paires de séquences, pour créer une matrice de proximité, et Nicholas créa un code HTML et Javascript pour permettre de changer les noms des séquences ainsi que pour avoir un tab options à droite et pour choisir la taille de l'ADN à générer aléatoirement. Dorénavant, la dernière étape importante du projet c'était l'arbre, donc Isabella regarda en ligne comment convertir le résultat dans la console en image, ou comment le visualiser, ainsi que comment faire un arbre phylogénétique en python, et Alek continua de regarder comment aligner des séquences. Après cela, il fallait juste réussir à tout lier, pour que tout fonctionne, donc Isabella a fini le CSS avec l'animation de la double hélice d'ADN.

Liste des tests à réaliser:

- Test avec fichier (séquences de protéines): Déposer le fichier séquences.fasta et exécuter le programme
- Test avec exemple.fasta
- Tester par entrée textuelle tout type de séquences

Même si on est très content avec le résultat, il y avait bien sûr quelques changements qu'on pouvait incorporer pour améliorer le tout. L'une des idées d'amélioration était de faire en sorte qu'on puisse implémenter des algorithmes sans l'aide d'un logiciel externe, juste directement sur le site web. De plus, on voulait aussi pouvoir créer une base de données qui puisse répertorier un grand nombre de séquences, comme ça l'utilisateur pourra sélectionner ces séquences comme options, ce qui rendrait l'UX beaucoup plus agréable. Avec cela on aurait aimé importer un fichier fasta, ce qui aiderait à stocker des séquences biologiques. Une autre amélioration qu'on aurait aimé faire c'est d'incorporer un tableau de comparaison pour chaque indice de proximité entre les séquences, en format 2 par 2. Finalement, la plus grande amélioration qu'on aurait du faire, c'est rendre le code plus compatible avec les Macs, parce que le fichier clustal n'est pas très très compatible avec tout ce qui est apparu après les macbooks catalina.

Compétences personnelles développées :

→ Isabella:

Comprendre le code avec plus de facilité, et comprendre qu'est-ce qui est lié à quoi, ainsi qu'apprendre comment faire de l'animation, et à nouveau comprendre qu'est ce qui fait fonctionner quoi.

→ Alek:

Travailler en équipe (organisation/ coordination), connaissances en biologie + algorithmes en bioinformatique.

→ Nicholas:

La mise en page avec html, css, et utiliser javascript pour rendre la page dynamique. Utilisation d'un logiciel externe pour de la bioinformatique en lisant des séquences fasta.

Schéma explicatif fonctionnement site web:

