

Rapport de projet

Alignement de séquences

GenoAligner



Groupe AR1_A:

- Ben Abdallah Montassar 'Info 5'
- Bouzid Lyna 'Info 4'
- Mohamed Halim Nafyssata 'Info 5'
- Mkouboi Mounia 'Info 4'
- Nur Akbas Aleysa 'Info 4'

2021/2022

Sommaire

Introduction	3
Partie I: Présentation du	
projet	3
Partie II: Organisation du groupe	
••••••	5
Partie III: Guide du	
programmeur	6
1-Fonctionnalités	6
2-Classes et fonctions	7
3-Problèmes rencontrés	10
Conclusion	11

Introduction

Dans le cadre de notre seconde année en Licence Informatique Générale, il nous a été demandé un projet de programmation d'un semestre de durée, afin de mettre en pratique nos connaissances et nos compétences en Programmation orientée objet ainsi qu'en Langage Java.

Ayant déjà eu l'occasion de réaliser un jeu et un site web, il nous est paru intéressant de nous diversifier en optant pour le sujet AR1 qui allie l'informatique et la biologie : L'alignement séquentiel.

Ainsi l'objectif est de développer un programme qui fait un alignement global maximal de deux séquences de nucléotides selon l'algorithme d'alignement de 'Needleman-Wunsch', en mettant à disposition des utilisateurs, une interface graphique qui se veut intuitive et dont la manipulation est agréable.

Pour mener à bien ce projet, nous nous sommes appuyés sur l'énoncé du sujet en guise de cahier des charges. Cependant, nous avons également fait appel à notre créativité et notre esprit d'équipe dans la réalisation de 'GENOaligner' que nous détaillerons dans ce qui suit.

Présentation du projet

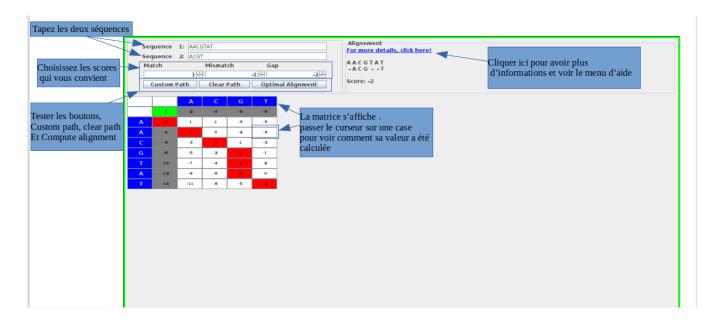
L'Alignement de séquences consiste à comparer deux séquences d'ADN pour évaluer un score optimal de similarité entre ces deux séquences à l'aide de l'algorithme d'alignement de Needleman-Wunsch. Cette technique est couramment utilisée par les bio-informaticiens pour faire un alignement globale maximale de deux séquences génomiques afin de ressortir les régions homologues.

Notre interface, le 'GenoAligner', consiste à aligner deux séquences saisies par l'utilisateur.

Lorsque le programme est lancé, nous sommes sur la page d'accueil de l'interface.



Une fois le bouton 'let's start' est cliqué, la fenêtre principale du projet s'ouvre. L'utilisateur pourra ensuite taper les séquences qu'il veut aligner et on aura une matrice qui s'affiche.



Organisation du groupe

Dans la conduite de ce projet, nous avons eu recours à la méthode Scrum, qui au total était composée de 5 sprints. Au bout de chaque sprint, nous organisions des réunions parfois en présentiel, d'autres en distanciel, afin de faire le point sur ce qui a été fait, ce qu'il reste à faire et pour redistribuer ou attribuer des tâches aux membres du groupe.

Pour garantir l'avancement du projet, il a fallu apprendre à communiquer, consulter l'autre pour des choix techniques, demander de l'aide en cas de besoin... WhatsApp a été notre principale plateforme d'échange, Nos réunions ponctuelles avec notre encadrante de projet nous permettaient aussi de garder le fil afin d'avancer dans la bonne direction.

Notre outil principal fut Gitlab, nous pouvions travailler simultanément sur des tâches différentes et ce dans des branches spécifiques à chaque fonctionnalité. Pourtant, la prise en main de cet outil ne fût pas évidente au début, ce qui générait des conflits, mais très vite nous apprenions de nos erreurs!

Notre branche principale est « develop », pour y ajouter du code il fallait un merge request et donc tester le bon fonctionnement du code avant de le merge dans develop, toutes les autres branches nous servaient de brouillon avant de tout mettre au propre dans develop.

Guide du programmeur

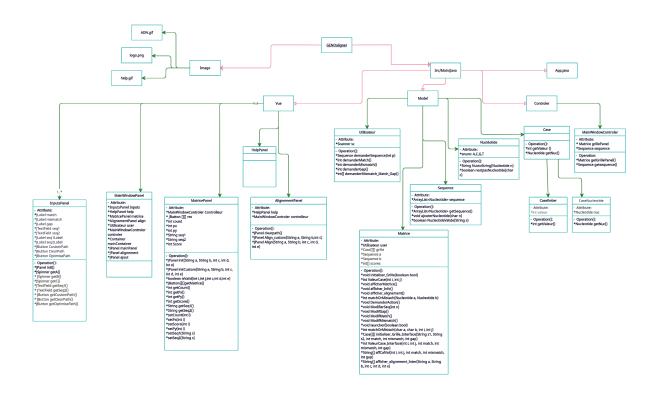
1. Fonctionnalités:

Dans cette partie, on va parler des fonctionnalités implémentées:

- → Une fenêtre d'accueil animé pour le projet.
- ➤ La possibilité d'aligner n'importe quelle paire de séquences de moins de 20 nucléotides.
- ➤ La possibilité de changer tous les scores: gap, match et mismatch avec les mises à jour nécessaires de la vue.
- ➤ L'affichage de l'alignement des deux séquences et du score final obtenu.
- ⇒ L'affichage de la matrice.
- → Le bouton "Compute Optimal Alignment" qui permet de mettre en évidence le chemin optimal avec l'alignement des deux séquences et le score obtenu.
- → Le bouton "Clear Path" qui permet de faire disparaître le chemin optimal de la matrice.
- → Le bouton "Custom Path" qui permet à l'utilisateur de choisir son propre chemin.
- → Une fenêtre qui s'affiche quand le curseur survole une case de la matrice pour expliquer comment la valeur de cette case a été calculée.
- → Une fenêtre d'aide d'aide qui donne quelques informations à l'utilisateur sur l'alignement global des séquences.
- → Des fenêtres d'erreurs qui s'affichent si une lettre tapée ne correspond pas à un nucléotide valide (A, C, G ou T) ou si la longueur de la séquence dépasse 20 caractères et encore quand l'utilisateur construit son propre chemin et choisit une case non valide.
- → Donner à l'utilisateur le choix de lancer le projet en mode graphique ou textuel.
- ➤ La possibilité de changer le gap, match et mismatch scores et les deux séquences en mode textuel.

2. Classes et fonctions:

Voici l'architecture de `GENOaligner`



Pour une meilleure visibilité, nous vous invitons à voir le fichier diagramme.png dans le GENOaligner.

Notre projet est fait selon l'architecture MVC (Model-Vue-Controller). Pour faire la liaison entre le modèle et la vue,on a défini un contrôleur avec comme attribut une instance de la classe matrice qui représente la classe principale du modèle. Mais dans cette classe, on a été obligé de redéfinir les méthodes du modèle qu'on aura besoin pour notre interface vu que les types des paramètres sont différents (nucléotide et string). Par la suite, on a utilisé le contrôleur à chaque fois qu'on avait besoin d'une fonction du modèle pour agir sur la vue. Aussi, durant tout le projet, on a essayé d'appliquer toutes nos connaissances pour avoir le meilleur résultat possible.

Dans cette partie nous allons expliciter certaines parties du code:

a) La fonction: initialiser Grille:

Cette fonction est une fonction de la classe matrice représentant le modèle. Elle prend en paramètre une variable booléenne et selon la valeur de vérité de cette variable la fonction va initialiser la matrice en demandant à l'utilisateur de nouveau les séquences à aligner ou c'est en gardant les mêmes séquences mais avec des scores(match, mismatch ou gap) qui ont été modifié.

```
public void initialiser_Grille(boolean bool) {
     if(bool) {
          this.a = user.demanderSequence( p: 1);
          this.b = user.demanderSequence( p: 2);
          this.scores = user.demanderMismatch_Match_Gap();
     this.grille = new Case[a.getSequence().size() + 2][b.getSequence().size() + 2];
     for (int \underline{a} = 2; \underline{a} < \text{this.a.getSequence}().size() + 2; <math>\underline{a} + +) {
          this.grille[a][0] = new CaseNucleotide(this.a.getSequence().get(a - 2));
     for (int b = 2; b < this.b.getSequence().size() + 2; <math>b++) {
          this.grille[0][\underline{b}] = new CaseNucleotide(this.b.getSequence().get(\underline{b} - 2));
     this.grille[1][1] = new CaseEntier( val: 0);
     for (int \underline{c} = 1; \underline{c} < \text{this.a.getSequence}().size() + 2; <math>\underline{c}++) {
     for (int d = 1; d < this.b.getSequence().size() + 2; d++) {</pre>
     for (int \underline{n} = 2; \underline{n} < \text{this.b.getSequence}().size() + 2; <math>\underline{n}++) {
          for (int m = 2; m < this.a.getSequence().size() + 2; m++) {</pre>
               this.grille[\underline{m}][\underline{n}] = \text{new CaseEntier(ValeurCase}(\underline{m}, \underline{n}));
```

b) La fonction: valeurCase:

Cette fonction fait aussi partie de la classe matrice. Elle prend deux paramètres entier i et j et permet de calculer la valeur d'une case situé à la position (i,j).

```
public int ValeurCase(int i, int j) {
   int a = (this.grille[i - 1][j - 1]).getValeur();
   if ((this.grille[0][j]).getNuc().equals((this.grille[i][0]).getNuc())) {
      a += this.scores[0];
   } else {
      a += this.scores[1];
   }
   int b = (this.grille[i - 1][j]).getValeur() + this.scores[2];
   int c = (this.grille[i][j - 1]).getValeur() + this.scores[2];
   return Math.max(Math.max(a, b), c);
}
```

c) La fonction: init:

Cette fonction est située dans la classe matricePanel. Elle retourne un JPanel qui contient la matrice remplie selon deux paramètres string ,représentant les séquences, passés en paramètres.

```
public JPanel init(String a, String b, int c, int d, int e) {
    this.res = new JButton[a.length() + 2][b.length() + 2];
    JPanel droite=new JPanel();

//droite.setPreferredSize(new Dimension(20 * (a.length() + 2), 20 * (b.length() + 2)));

droite.setLayout(new GridLayout( rows a.length() + 2, cols b.length() + 2));

//TODO: initiliaser le texte des Jbuttons pour éviter d'avoir NullPointerException par la suite

for (int i = 0; i < a.length() + 2; i++) {
        for (int i = 0; j < b.length() + 2; j++) {
            res[i][i] = new JButton( text: " ");
            res[i][i].setForeground(Color.BLACK);
            res[i][i].setBackground(Color.BLACK);
            res[i][i].setForeground(Color.WHITE);
        }
    }

//TODO: remplir la lêre colonne de la matrice

for (int i = 2; i < a.length() + 2; i++) {
        res[i][i].setForeground(Color.white);
        res[i][i].setBackground(Color.BLUE);

        res[i][i].setForeground(Color.BLUE);

        res[i][i].setText(" " + String.valueOf(a.charAt(i - 2)).toUpperCase());
    }

//TODO: remplir la lêre Ligne

for (int i = 2; i < b.length() + 2; i++) {
        res[i][i].setText(" " + b.charAt(i - 2));
        res[i][i].setForeground(Color.white);
        res[i][i].setForeground(Color.white);
    res[i][i].setBackground(Color.white);
    res[i][i].setBackground(Color.white);
    res[i][i].setBackground(Color.white);
        res[i][i].setBackground(Color.white);
        res[i][i].setBackground(Color.white);
        res[i][i].setBackground(Color.white);
        res[i][i].setBackground(Color.white);
        res[i][i].setBackground(Color.white);
        res[i][i].setBackground(Color.white);
        res[i][i].setBackground(Color.white);
        res[i][i].setBackground(Color.white);
        res[i][i].setBackground(Color.white);
        res[i][i].setBackground(Color.white);
        res[i][i].setBackground(Color.white);
        res[i][i].setBackground(Color.white);
        res[i][i].setBackground(Color.white);
        res[i][i].setBackground(Color.white);
        res
```

d) La fonction: Align:

Cette fonction est dans la classe AlignementPanel. Elle retourne un JPanel qui contient l'alignement des séquences passées en paramètre.

```
public JPanel Align(String a, String b, int c, int d, int e){
    Case[][] g=this.controlleur.getGrillePanel().initialiser_Grille_Interface(a, b, c, d, e);
    String[]x=controlleur.getGrillePanel().afficher_alignement_Inter(a,b,c,d,e);
    String m="";
    String m="";
    String n="";
    for (int i=x[0].length()-1;i>=0;i--) {
        if (String.valveOf(x[0].charAt(i)).equals("-")) {
            m += " " + String.valveOf(x[0].charAt(i)).toUpperCase()+" ";
        } else {
            m += String.valveOf(x[0].charAt(i)).toUpperCase()+" ";
        }
    }
    for(int j=x[1].length()-1;j>=0;j--) {
        if (String.valveOf(x[1].charAt(j)).toUpperCase()+" ";
        } else {
            n += " " + String.valveOf(x[1].charAt(j)).toUpperCase()+" ";
        } else {
            n += String.valveOf(x[1].charAt(j)).toUpperCase()+" ";
        }
    }
    JTextArea labelArea = new JTextArea("\n"+m+"\n"+n+"\n"+x[2]);
    labelArea.setEditable(false);
    labelArea.setEditable(false);
    JPanel AL=new JPanel();
```

3. Problèmes rencontrés:

Le problème majeur qu'on a eu c'était la différence d'affichage de la matrice entre les utilisateurs de Mac et les utilisateurs de Linux. Ceux qui avaient un Mac n'ont pas eu la même matrice tandis que le code est exactement pareil pour tout le monde.

Conclusion

Pour conclure, ce rapport relate les différentes étapes, choix techniques, problèmes par lesquels 'GENOaLigner' a vu le jour, ce projet nous a permis d'appliquer les connaissances que nous avons acquises durant nos deux premières années universitaires, telles que la modélisation, le langage JAVA, la gestion de projet mais aussi la communication et une nouvelle culture générale en biologie.