Djabali Yacine

Parmentier Raphaël

Vencic Jean

M1 BIBS

Rapport Etude de cas

## I - Contexte

Tout d’abord, une analyse du fichier format .gff3 nous a permis de visualiser comment est organisé un génome de façon numérique dans ce format.

En effet, nous avons remarqué que les informations étaient stockées sous forme de tableau (avec des “\t” comme délimiteur), avec en ligne les différentes régions qui compose le génome et en colonnes des informations caractérisant ces régions.

On distingue 9 colonnes :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Seq ID | Source | Feature | Start | End | Score | Strand | Frame | Attributes |

En s'intéressant plus en détail aux lignes du fichier, nous avons pu remarquer une hiérarchie entre les régions.

Par exemple : une région qui s’étend de x a y peut être caractérisé par un type particulier : un gène, un pseudogène, régions répétées, etc ainsi que d’autres informations associées.

Par la suite, en dessous de celle-ci on va retrouver une région comportant un intervalle inclus ou égal au précédent. Ce nouveau type de région, que nous appellerons “sub region” présente de nouvelles informations et des types spécifiques : CDS, tRNA, rRNA, …. Par la suite on rencontrera une nouvelle ligne avec un intervalle non inclus et non chevauchement avec les précédents marquant le début d’une nouvelle “région” suivient de ses sous-régions.

## II - Notre démarche

Nous nous sommes donc basé sur cette organisation hiérarchique afin de concevoir notre programme:

**PARSEUR :** Notre parseur récupère le fichier d’input et grâce à la séparation des colonnes par tabulation, le range dans une array liste. Les informations sur le génome seront stocké suivant la hiérarchie avec pour chaque région, sous-régions un objet propre qui sera instancié et cela au fur et à mesure de la lecture du fichier.

**INTRO**: Nous avons donc pensé à un système imbriqué d'agrégation d’objet. Au départ, l’idée de l’héritage nous semblait intéressante notamment entre les régions et les sous régions car elles portent des attributs en commun mais ayant besoin de représenter l’imbrication de ces objets entre eux, le stockage des objets de classes hiérarchiquement inférieur dans des listes à la façon d’une agrégation nous a paru plus pertinente.

**DÉTAILS DES CLASSES:**

GÉNOME : Génome contiendra des informations propres à l’intégralité du génome contenu dans le fichier d’input, ainsi que la liste exhaustive des régions, sub-regions et sub-sub regions qui le compose (agrégation = flèche sur le diagramme).

RÉGION : S’en suit une classe Région qui contiendra un lot d’informations précises : type, start, stop, stand, taille, e-val, info et la liste de ses sous-région. Les régions sont les éléments hiérarchiques les plus haut qui composent le génome. Elles sont reconnaissables par leur type spécifique (gene, pseudogene, repeated region, …) et le fait que leurs bornes ne sont jamais chevauchantes ou incluses entre elles. Chaque région contient des sous-régions propres dont les bornes sont incluses dans celle de la région considérée.

SUB-REGION : Cette classe représente les sous-régions. Elles contiennent quelques informations supplémentaires que la classe région comme la phase de lecture, le cadre de lecture et la liste de sub-sub-region associées.

SUB-SUB-REGION : Cette classe a été rajoutée pour les exons qui sont d’un niveau hiérarchique inférieur aux sous-régions et ne peuvent donc pas être considérées comme telles. Elles détiennent les mêmes attributs que les sub-region sauf la liste de sous-sous régions incluses bien sûr.

## III-Interface graphique

Pour notre projet il va nous falloir une interface graphique, n’ayant pas trop de connaissances sur cet aspect nous ne sommes pas encore certains de la finalisation de celle ci.

* L’affichage devrait-il modulable sur un intervalle précis ou sur tout le génome ?
* L’échelle risque d’être dur à définir, mais une fois celle ci mise au point il ne devrait pas être trop dur d’afficher des boutons sur un axe horizontale pour afficher nos régions.
* Dans l’idéal nous aimerions que les boutons est une application (ex : quand on appuie sur une régions les information concernant celle-ci apparaissent dans une bulle), cela me semble plus dur à mettre en oeuvre.

Pour la structure de l’interface graphique, nous imaginons deux classe, une fenêtre et un dessin. Nous n’avons pas jugé bon de les mettre dans le diagramme UML car nous n’en sommes qu’à la phase de prototypage de la structure de l’interface.

## IV - Diagramme UML

## 