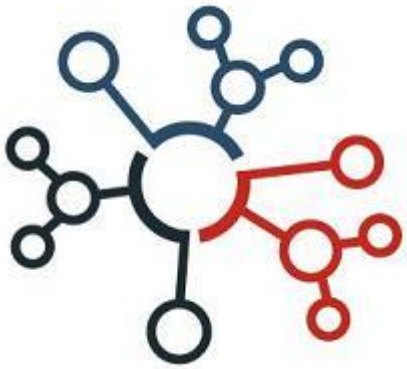
11/2018



Année universitaire	: 2018/2019
Université	: Ecole National des sciences appliquées
Encadre par	: Pr. AMNAI
Réaliser par	: ELBAGHAZAOUI Bahaa Eddine

1 Introduction

UniProt est une base de données de séquences de protéines. Son nom dérive de la contraction de Universal Protéine Resource (base de données universelle de protéines). C'est une base de données ouverte, stable et accessible en ligne, elle est issue de la consolidation de l'ensemble des données produites par la communauté scientifique. UniProt est une base annotée, hiérarchisée où chaque séquence est accompagnée d'un ensemble riche de métadonnées et de liens vers de nombreuses autres bases de données : bibliographiques, phylogénétiques, nucléotidiques1... Outre la séquence en acides aminés des protéines, UniProt fournit des informations sur leur fonction et leur structure ainsi que des liens vers d'autres bases de données.

UniProt combine les données des bases Swiss-Prot, TrEMBL et Protein Information Resource (PIR) et est mise à jour régulièrement. Ses données reposent entre autres sur le serveur ExPASy de l'Institut suisse de bioinformatique et celui de l'EBI. Ces serveurs proposent en particulier la recherche de séquences homologues dans la base au moyen d'outils d'alignement de séquences comme FASTA ou BLAST.

Content

2 Accès programmatique

Uniprot est une massive base de données des protéines, pour faciliter l'échange avec les développeurs. UniProt fournit plusieurs interfaces de programmation d'application (API) pour interroger et accéder à ses données par programmation :

- UniProt website REST API
- Proteins REST API
- UniProt SPARQL API
- UniProt Java API

Dans notre cas, on va travailler par « UniProt Java API » qui contient des méthodes déjà implémente dans le code source.

2.1 Code Source

Pour préparer l'environnement de développement on est besoin de :

- Java 8 SDK installation.
- Outil pour coder et compiler « notre cas : Intellij Idea »

Uniprot offre le dossier ci-dessus qui contient les classe qui facilite la communication avec la base de donneés avec la description d'importer les jars au projet « https://www.ebi.ac.uk/uniprot/japi/ ».

	Size (MB)	Architecture	Type	Content
uniprot-japi-client.zip	27	Platform-Independent	.zip	UniProtJAPI client Jar All required libraries Documentation Examples

Dans notre projet, après l'installation et l'importation on trouve des classe qui contient des méthodes qui facilitent la recherche sur une information qui va nous intéresse, les classes commence soit par :

- UniProt[.] : cherche par les attributs des protéines.
- UniRef [.] : cherche par clusters des protéines
- UniParc[.] : cherche par base de données des protéines.

Uniprot affect les données d'une protéine à une classe s'appelle « UniProtEntry » C'est pour cela on crée une classe s'appelle Protéine qui contient :

- Entry : objet de la classe UniProtEntry .
- Neighbors : table de hashage contient un voisin comme clé et distance Jaccard comme valeur .

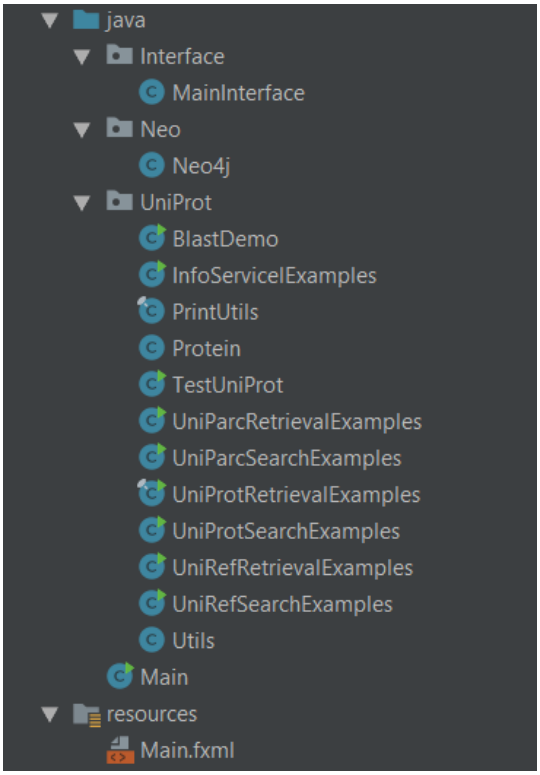
2.2 Architecture Code

- Interface

Un package qui contient le controller de l'interface « Main.fxml »

- Neo

Contient la classe « Neo4j » qui permet d'affiche le résultat dans le programme Neo4j Desktop.



2.2.1 Neo4j

Apres l'installation du Neo4j, il existe un utilisateur par défaut « neo4j,password » qui est déjà configurer comme administrateur.

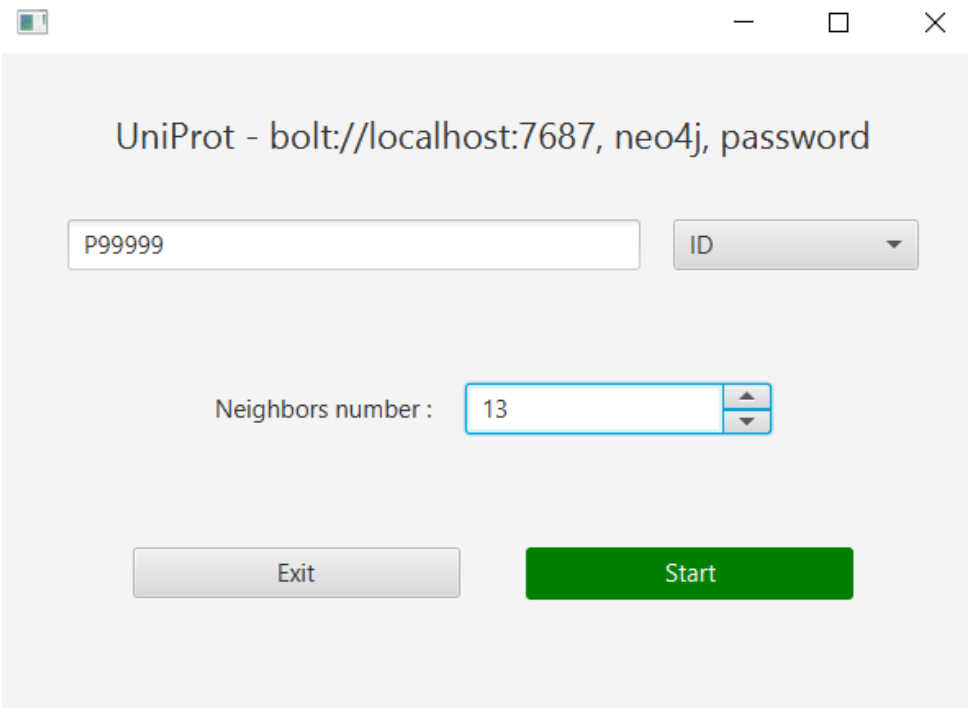
Au cas d'erreur il faut créer un autre utilisateur et donne « admin » comme rôle à ce nouveau ou bien vérifie le port « bolt : 7687 ».

3 Procédure de Réalisation

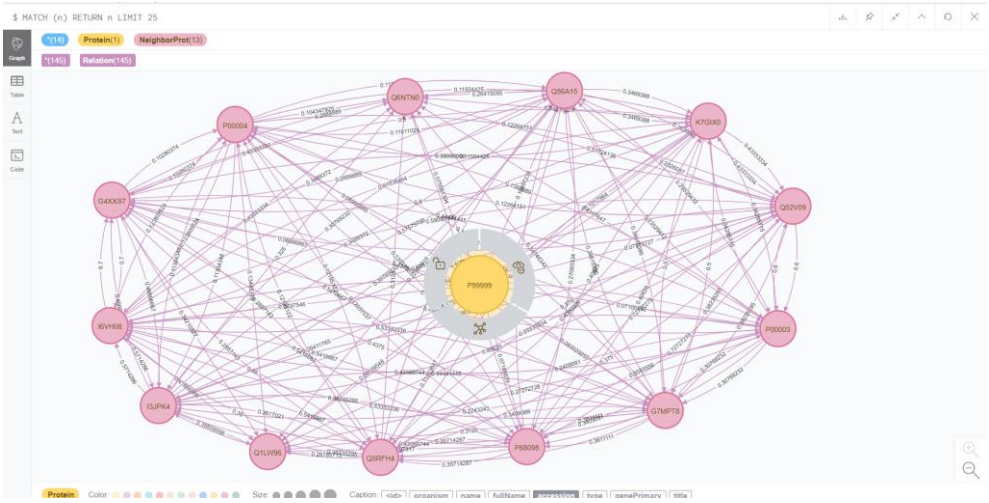
Dans notre projet, Nous venons de cherche dans UniProtKB qui contient les informations sur les protéines, tous d'abord on cherche par « ID, Name, FullName … ». le programme nous répond par tous les valeurs des attributs de ces protéines, ensuite on cherche par le cluster qui contient cette protéine. Comme réponse, on obtient les id de chaque protéine appartient à ce cluster. Par suite, on recherche les informations de chaque protéine de cette cluster.

La distance de Jaccard se calcule d'après la balise « dbReference » qui signifie les « Sequence databases » qui contient la protéine. On faire une comparaissant entre la protéine puis on calcule l'indice de Jaccard.

4 Exécution



On obtient comme résultat :



5 Contraintes

D'après la recherche qu'on a effectuée, on trouve que la réponse d'après uniprot contient toutes les informations du protéine, cluster …, cela implique que la recherche sera très longue et la réponse contient des informations seront inutile sur certains cas. Chaque protéine appartient à un seul cluster.

```
<gene>
      <name type="primary">CYCS</name>
      <name type="synonym">CYC</name>
</gene>
```