Base de données NoSQL

Rapport de projet

LEBRETON Baptiste
RIVIERE Mickael
SAMSON Ulrik

Décembre 2022

Table des matières

1 Objectif

L'objectif de ce projet est d'obtenir un outil permettant d'interroger deux bases de données NoSQL (MongoDB et Neo4J) sur les protéines. Une interface Web doit être mise à disposition de l'utilisateur afin qu'il puisse effectuer ses recherches sur le nom ou l'identifiant de la protéine. La recherche aboutira sur un affichage des informations recueilli dans les bases de données, ainsi qu'un affichage des voisins de ces protéines se basant sur la similarité de jacquard de la valeur de leurs champs "interpro".

2 Architecture

Il a été décidé d'utiliser le Framework Flash (Python). Ce Framework est versatile et permet d'utiliser un grand nombre de librairies compatible avec les bases de données NoSQL. Dans notre cas, la liaison avec MongoDB se fera via la librairie PyMongo. Elle permet notamment de faire les requêtes sur la base de données et ainsi récupérer l'ensemble des informations d'une protéine. Pour Neo4J, la librairie Neo4J permet de créer les relations et dans un second temps, Neovis est utilisé pour récupérer les informations et pour l'affichage de la protéine et ses voisines. La **figure 1** permet de visualiser notre architecture et les différentes interactions qu'ont chacune des parties de notre système entre elles.

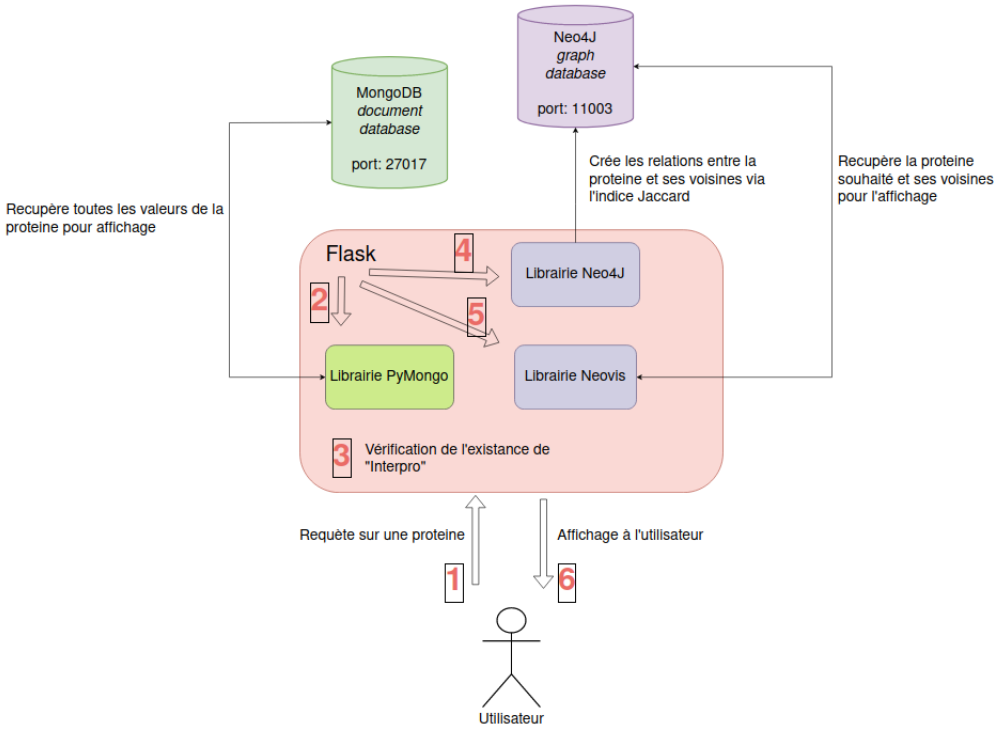

FIGURE 1 – Schéma de l'architecture globale

3 La base de données MongoDB

La base de données MongoDB est une base de données orientée documents. Elle permet ainsi d'obtenir toutes les informations telles que le nom, l'identifiant ou encore la description d'une protéine. On peut voir sur la **figure 2** que la base est bien implémentée dans notre système et qu'elle contient la liste de toutes les protéines nécessaires à ce projet.
Son interrogation est effectuée en back-end de notre solution grâce à la librairie *pymongo* de python. MongoDB permet d'effectuer des requêtes avec un temps de réponse très faible. Dans notre système, son utilisation est donc principalement pour lister les informations d'une protéine.

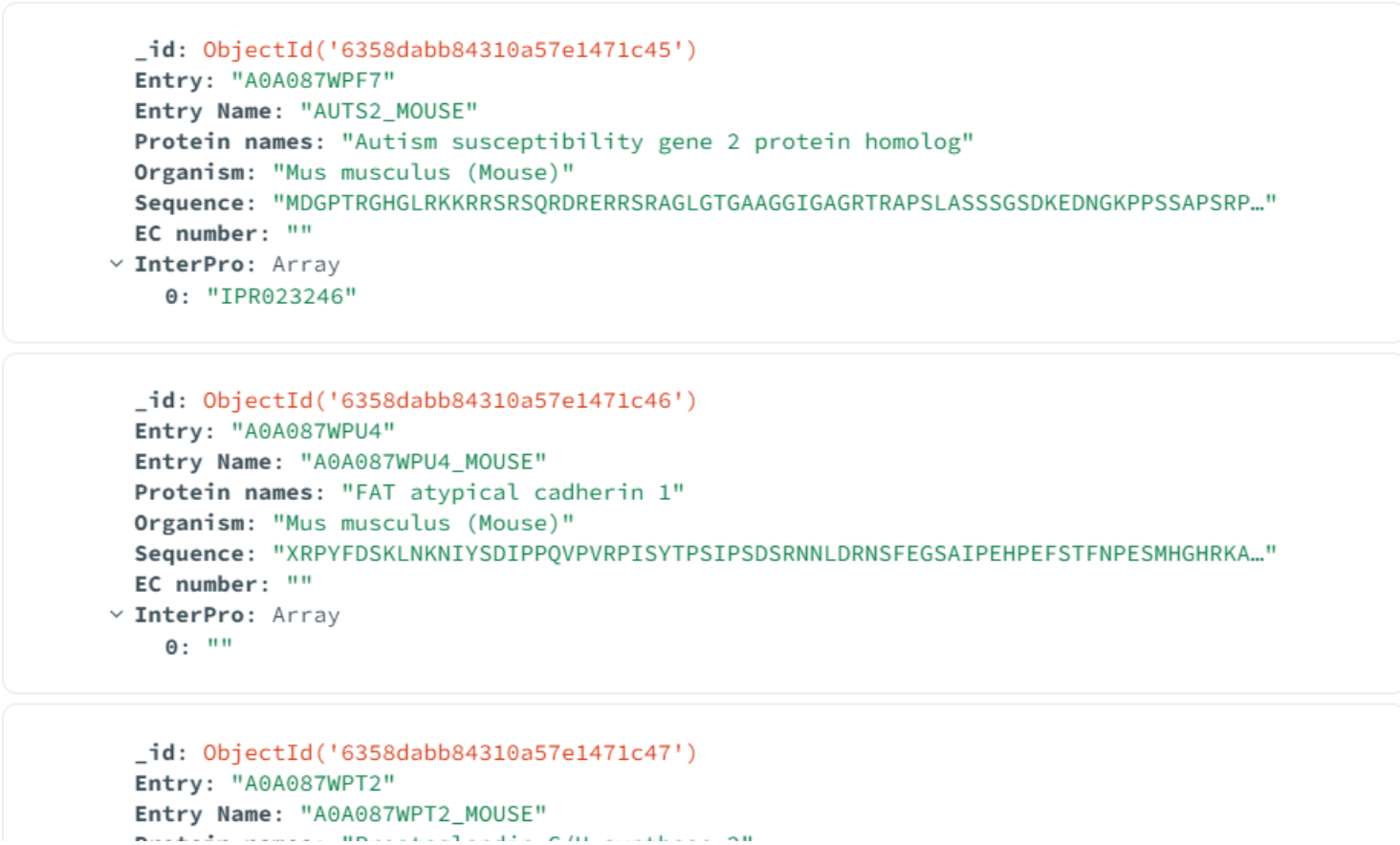

FIGURE 2 – Base de données MongoDB

4 La base de données Neo4j

La base de données Neo4J est une base de données orientée graphe. Elle permet ainsi d'effectuer des opérations plus complexes. Cependant, cela influera sur le temps de réponse des requêtes qui peut être très important. Dans notre système, son utilisation est donc principalement pour obtenir une représentation graphique des protéines et leur voisin.
Sur la **figure 3**, on voit que la base est présente dans notre système et permet de représenter, les protéines présentent ainsi que leur relation de voisinage sous la forme d'un graphe relationnel. Neo4J effectuera également les opérations liées au calcul de la similarité de jacquard de la protéine voulue à chaque requête. Son interrogation est effectuée en back-end de notre solution grâce à la librairie *neo4j* de python. Afin de faciliter ces requêtes, quelques modifications ont été apportées à la base de données initiale. En effet, les "interpro" qui était originellement des chaines de caractères, ont été remplacé par une liste de chaines de caractères, facilitant ainsi le travail de recherche sur ces dernières.

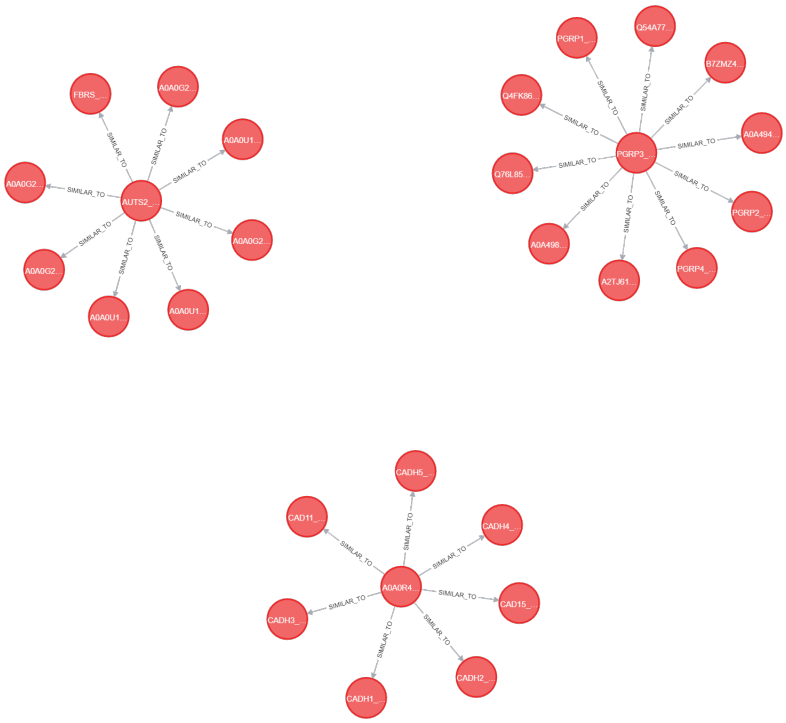

FIGURE 3 – Base de données Neo4J

5 L'interface utilisateur

Afin de pouvoir visualiser les éléments présenter dans les parties précédentes, notre solution dispose d'une application web utilisant Python (Framework Flask) pour le back-end, Bootstrap pour le front-end et NeoViz pour l'affichage du graphe. Sur la **figure 4**, on peut voir un exemple d'affichage lors d'une requête sur les bases données.

Ainsi, lors de l'accès à notre serveur Web, une page d'accueil affichant diverses statistiques sur la base de données est présente. Une barre de recherche disposant d'options afin de changer le paramètre de recherche et la valeur de la similarité de jacquard est disponible pour interroger les bases de données. La requête est donc envoyée aux deux bases de données permettant ainsi une covisualisations des résultats. L'affichage du résultat de la requête sur la base **MongoDB** se fait sous la forme d'une liste d'informations sur la protéine recherchée. Concernant la base **Neo4J**, le résultat est sous la forme d'un graphe permettant de visualiser le nœud recherché et ses voisins. La valeur du coefficient de Jaccard est mise en évidence via l'épaisseur des liens entre les nœuds.

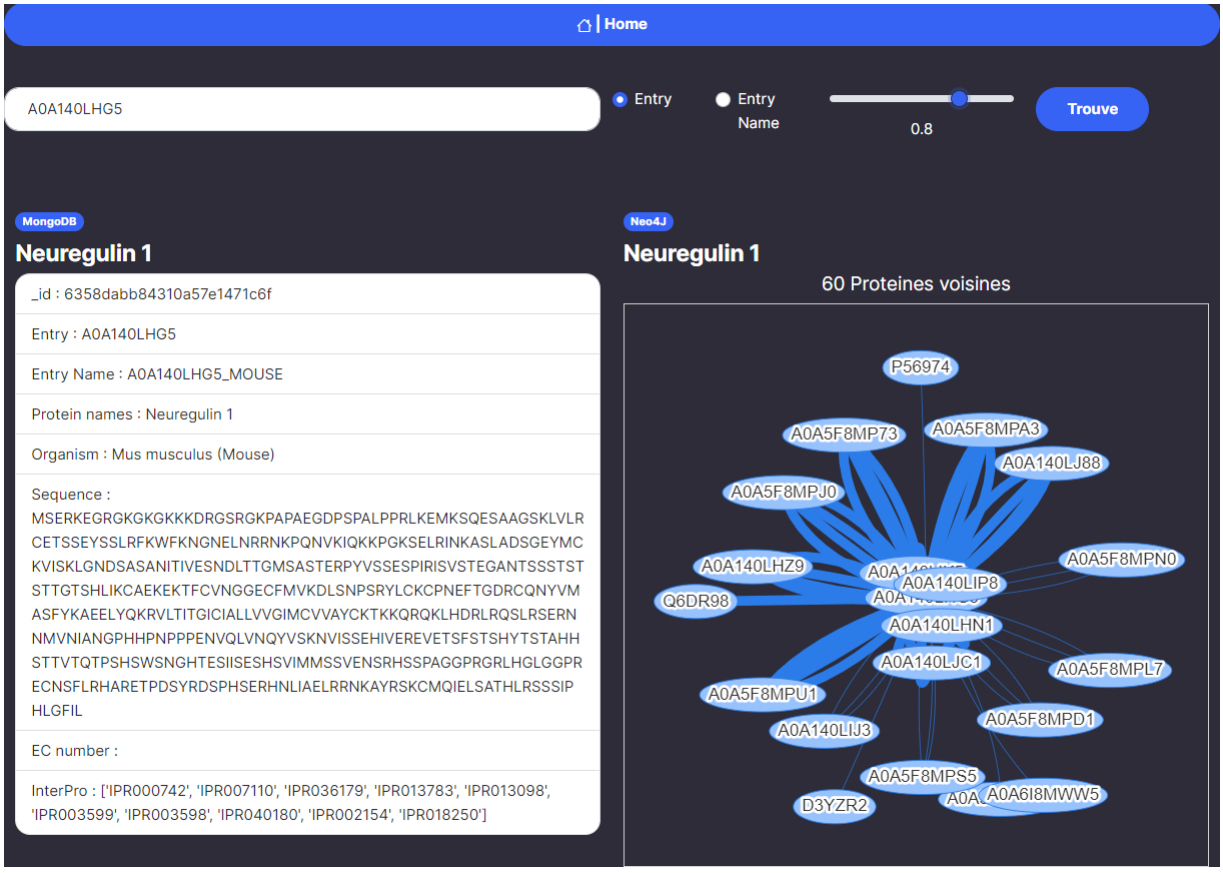

FIGURE 4 – Recherche d'une protéine avec Entry = "A0A140LHG5" et une similarité de jaccard supérieur à 0.8