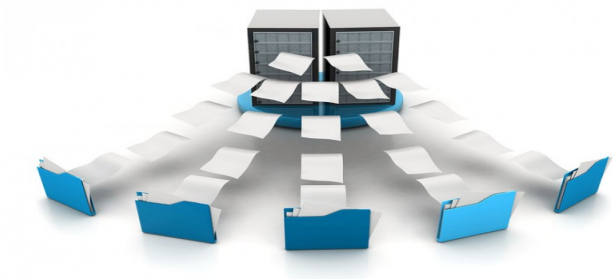
Rapport de stage Master 2 Bio-informatique : parcours analyse de données génomiques

Université de Rennes 1
UFR Science De La Vie

Amélioration d'un outil d'intégration de base de données biologiques liée à Elasticsearch



Sous la direction de M Régis Ongaro-Carcy et du Professeur agrée Arnaud DROIT

(Arnaud Droit Lab.)

CERIL Hans – Master 2 –Université de Rennes 1- 2017/2018

UMR CNRS 6290
Institut de Génétique et Développement de Rennes (IGDR)

Annabelle MONNIER
Equipe Génomique Fonctionnelle Intégrée et Biomarqueurs

UNIVERSITÉ DE RENNES 1

ENGAGEMENT DE NON PLAGIAT

Je, soussigné(e) .....
étudiant(e) en.....
déclare être pleinement informé que le plagiat de documents ou d'une partie de document publiés sur toute forme de support, y comprit l'internet, constitue une violation des droits d'auteur ainsi qu'une fraude caractérisée.

En conséquence, je m'engage à citer toutes les sources que j'ai utilisées pour la rédaction de ce document.

Date :

Signature :

Document à compléter de manière manuscrite et à insérer obligatoirement en première page du rapport de stage.

Laboratoire de Génomique Médicale
BMT-HC - CHU Pontchaillou

2 rue Henri le Guilloux
35033 Rennes Cedex
FRANCE

Annabelle MONNIER
annabelle.monnier@univ-rennes1.fr
TÉL. 33 (0)2 99 28 92 54

Remerciements

Au terme de ce travail, j'adresse mes remerciements les plus sincères à mon encadrant Régis-ONGARO-CARCY de m'avoir permis de bénéficier de son savoir et de ces compétences tout le long de mon stage.

Un grand merci au Dr Arnaud DROIT, de m'avoir offert cette chance et cette opportunité de venir effectuer mon stage au sein de son équipe.

Je ne manquerai pas d'oublier mes parents pour leurs disponibilités, leurs précieux conseils et leurs soutiens tout au long de cette année d'étude, même dans les moments les plus difficiles.

Description des abréviations

ETL :

Sxcscs

Csds

Czzcs

Cscscs

Cscs

Cscscs

Cscscs

Cscscs

cscscsc

Résumé

Abstract

Table des matières

I. Introduction

1. BIG DATA : l'évolution de le médecine moléculaire

Durant ces dernières années, le domaine de la médecine moléculaire a connu une forte transformation dû à une explosion des données par séquençage massif, porté notamment par l'avancée du projet « Human Genome »[1].

L'apparition du séquençage de nouvelle génération « NGS » a permis de révolutionner la recherche scientifique en réduisant le coût de production des données scientifiques. Le volume de toutes ces données constitue l'exemple le plus évident de l'avancement des technologies de séquençage de nouvelles générations (NGS) [2].

En effet, le coût de séquençage a chuté plus rapidement de ce que prévoyait la « loi de Moore »[3] «Figure 1», cela démontre que, si cette production de données continue à augmenter à un coût fixe, il sera indispensable, dans les années à venir, d'avoir accès à suffisamment de mémoire afin de stocker toutes ces données[4].

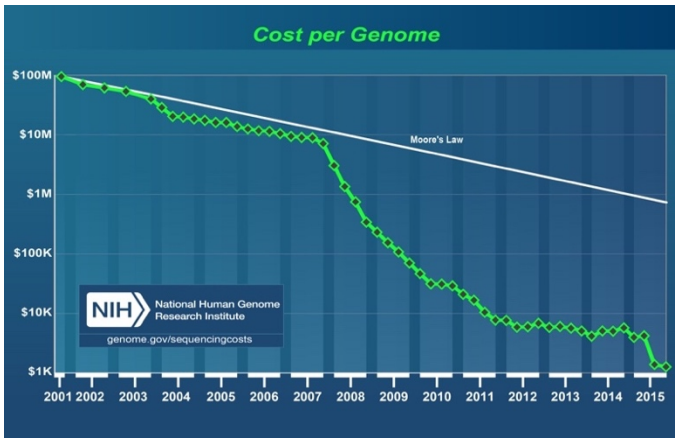


Figure 1 : Graphe démontrant la diminution du coût du séquençage au cours du temps

Cette capacité à saisir toute cette masse d'informations pouvant aller de la chimie médicinale à la pharmaco-génomique a permis d'inaugurer une nouvelle ère pour la recherche biomédicale.

Parmi toutes ces données à haute résolution, on peut ainsi générer de nouveaux profils moléculaires à travers un large éventail de paramètres qu'ils soient moléculaire (génome, transcriptome, protéome ou métabolome), environnementaux ou comportementaux[5].

2. Forte croissance du nombre de bases de données biologiques

Parallèlement à cette accumulation de données biologiques à grande échelle, le nombre de bases de données biologique développées pour gérer cette masse d'information, augmente de plus en plus[6].

Ces bases de données sont donc devenues des outils primordiaux pour la recherche, car elles fournissent un accès aux connaissances pouvant être indispensable à la planification des expériences futures et à la découverte de nouvelles connaissances par exploration des données[7].

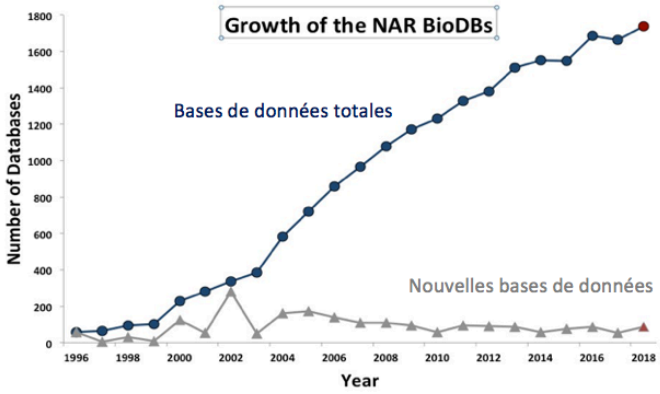


Figure 2: Graphe démontrant l'évolution des bases de données au cours du temps

En effet, en raison de leurs importances cruciales, le nombre de bases de données biologiques publiques ne cesse d'augmenter chaque année. Selon le site web « Nucleic Acids Research », on recense en 2018 un total de 1737 bases de données en biologie moléculaire par rapport à deux bases de données en 1980[6].

3. Application des « Warehouses » (Entrepôt de données)

Toute cette grande quantité d'informations générées en science de la vie est généralement dispersée dans plusieurs bases de données et répertoires différents. Parmi les méthodes « d'intégration » (processus par lesquels les données provenant de différentes parties du système d'information sont déplacées, combinées et consolidées) de données les plus communes en bioinformatique, on retrouve celles visant à intégrer ces données provenant de différentes sources de données biologiques dans un espace de stockage commun[8].

En effet dans le passé, diverses approches d'intégration ont été développées, telles que « Atlas »[9] et « BioWarehouse » [10]. Le développement de ces systèmes a pour but d'intégrer des données biologiques hétérogènes. L'une des particularités de ces approches d'intégration est la mise en place de « Data Warehouses », afin d'assembler des données ciblées pour des analyses bio-informatiques et la découverte des relations scientifiques entre ces données.

Selon Inmon[11], un « Data Warehouses » est défini comme une collection de données intégrées, non-volatiles, variant dans le temps.

En résumé, ces entrepôts de données sont dédiés au stockage des données structurées non volatile provenant de différentes sources et conçu de manière à pouvoir analyser et visualiser les données. En plus d'une base de données relationnelle, il intègre généralement un outil d'extraction, de transformation et de chargement des données « ETL »[12].

Ce processus d'ETL constitue la pierre angulaire d'un entrepôt de données puisqu'il a pour but, d'extraire les données des systèmes sources, de maintenir la qualité des données, d'appliquer des règles standards de transformation, de les charger dans un entrepôt principal ou distribué et pour finir, de présenter les données sous diverses formes[13].

4. Sujet de stage et principale contribution

L'une des premières préoccupations du phénomène du « Big Data » est le volume d'informations auquel les chercheurs doivent faire face. Les systèmes de fichier *(façon de stocker les informations et de les organiser dans des fichiers)* et la fiabilité de stockage *(capacité de conserver les données pendant de nombreuses années)* représentent les principales difficultés rencontrées.

Le vrai défi aujourd'hui est d'accéder à ces données de manière efficace, en appliquant un parallélisme massif, non seulement pour le calcul, mais aussi pour le stockage[14].

Afin de pouvoir exploiter le potentiel de toutes ces ressources moléculaires, elles doivent être stockées, combinées et utilisées dans des nouvelles applications informatiques, afin d'extraire le plus d'informations que possible.

Ainsi, pour répondre à ces besoins, une partie des travaux du laboratoire d'Arnaud DROIT porte sur le développement des outils informatiques permettant l'intégration et la visualisation des données biologiques en relation avec la médecine moléculaire.

Un logiciel est actuellement en phase de production par un doctorant du laboratoire, Régis-ONGARO-CARCY. Cet outil web nommé « KIBIO » permet d'intégrer différentes banques de données, d'y accéder, de visualiser les données sous-jacentes et d'interconnecter les banques via les identifiants de référence internes.

Ce logiciel informatique se compose de trois outils reliés entre eux, ayant chacun un rôle précis :

- **Lucan** : ETL permettant d'extraire, transformer et charger les données dans un entrepôt de données (Elasticsearch).
- **Elasticsearch** : Serveur utilisé pour l'indexation et la recherche des données[5].
- **KIBI** : Utilisé pour la visualisation des données et l'interconnexion de celles-ci par l'intermédiaire d'ID de référence.

Figure 3 : schéma représentatif de KIBIO

Au cours des six mois de stage de fin d'année de master, mon travail s'est focalisé sur l'amélioration et l'intégration de nouvelles fonctionnalités de « LUCAN », plus précisément

sur l'intégration de plugins informatiques permettant d'extraire, de transformer et de charger les données en relation avec la médecine moléculaire dans Elasticsearch qui est une base de données NoSQL.

5. Déroulement

Indirectement, le stage a débuté dès le mois de Novembre 2017, du fait qu'il nous a été demandé par nos responsables de formation, de faire une présentation bibliographique de notre sujet ainsi que de la remise d'un résumé descriptif.

Le stage s'est déroulé comme suite :

- ✓ La mise en place de mon espace de travail (système d'exploitation et outils nécessaire),
- ✓ La recherche bibliographique plus approfondie,
- ✓ La compréhension du code source de Lucan,
- ✓ L'échange avec mon maître de stage des différentes bases de données à intégrer,
- ✓ Le développement et intégration des plugins informatiques dans Lucan,
- ✓ La compréhension du code source du portail d'accès à Kibio,
- ✓ L'ajout de nouvelles fonctionnalités sur le site web,
- ✓ La rédaction du rapport de stage,
- ✓ La présentation du sujet au sein de l'équipe.

Durant ces phases, j'ai eu l'opportunité d'assister à de nombreuses formations et séminaires de recherche tels que :

- ✓ Les séminaires de recherche des différents membres de l'équipe, qui m'a permis d'appréhender les différents thèmes du laboratoire.
- ✓ La journée « Génome QUEBEC : The futur of omics » présenté par :
  - Dr Brenda Andrews, PhD, CC, FRSC. Director of Donelly Centre for Cellular and Biomolecular Research University of Toronto.
  - Dr Michael Moran, PHD, Director of SPARC Biocentre of the university of Toronto and Hospital for Sick Children.
- ✓ La formation et délivrance d'une attestation « Analyse et visualisation de données en python » à l'université de Laval (Québec),
- ✓ La formation et délivrance d'une attestation « Mise en pratique de l'apprentissage automatique avec scikit-learn » à l'université de Laval (Québec).

Ces journées ont été une source de connaissance, tant sur le plan d'acquisition des notions, mais également sur le plan Humain.

II. Matériels et méthodes

1. Principe et fonctionnalité de Lucan

a. Présentation

LUCAN est un logiciel informatique, développé en python3.5 faisant intégralement partie de KIBIO. Il est pour le moment uniquement destiné à traiter les données en science moléculaire. La particularité de cet outil, c'est d'être un logiciel back-end de multitraitement personnalisé, prenant en compte les différentes fonction d'un ETL telles que :

- ✓ L'extraction des données provenant de différentes sources de données,
- ✓ La transformation des données,
- ✓ Le chargement des données dans Elasticsearch.

LUCAN est le cœur de KIBIO et de par son architecture complexe, il est composé de différents plugins informatiques ayant pour but d'effectuer les trois fonctions principales d'un ETL, citées précédemment. Chacun de ces plugins sera spécifique à une base de données, permettant en cas de problème de s'y référer immédiatement.

b. Les formats de données

Les formats de données désignent un ensemble structuré d'informations servant à définir ou décrire une ressource. On retrouve également, les métadonnées qui représentent les « données sur des données » permettant ainsi à un individu ou un ordinateur de comprendre les sens et l'organisation de celles-ci.

Les données provenant du domaine de la médecine moléculaire que nous avons intégrées dans LUCAN sont de type :

- ✓ **Structurées** : données réorganisées afin de mieux exploiter les informations. Les exemples de formats rencontrés sont :
  - CSV « Comma-Separated Values »
  - TSV « Tab-Separated Values »
- ✓ **Semi-structurées** : XML « Extensible-Markup Language » ou JSON « JavaScript Object Notation ». C'est une structure de fichier arborescente.
- ✓ **Non-structurées** : données représentées ou stockées sans format prédéfini : Txt « texte brut ».

c. Des sources de données à LUCAN

La première étape concernera la sélection des bases de données d'intérêt. Une étude bibliographique des bases de données sources a été effectuée. Son but est de savoir quelles peuvent être leurs contributions dans l'avancement de nos travaux de recherche.

L'analyse complète sur l'identification et la signification des attributs permettra en effet de :

- ✓ Faciliter l'intégration de celles-ci.
- ✓ Déterminer sa contribution en termes d'informations moléculaires.
- ✓ Déterminer ces relations avec les autres bases de données.

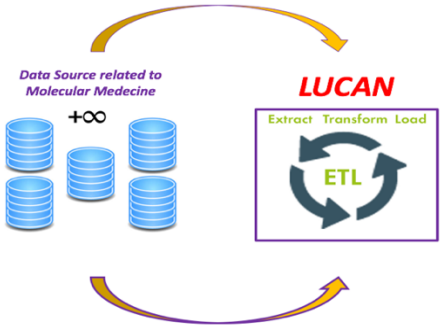


Figure 4 : représentation schématique de l'intégration des données dans LUCAN

Chaque base de données possède un ensemble distinct de caractéristiques qui doit être étudié, afin d'extraire et de transformer efficacement les données.

LUCAN doit intégrer des systèmes de gestions de base de données ayant des formats de structure de données différentes. C'est la raison pour laquelle, il sera important de bien se renseigner sur la façon dont chaque format de données doit être traité[15].

d. Transformation des données

Après le processus d'extraction, les données sont transformées dans un second temps pour matcher le schéma du Datawarehouse cible. Cette étape de transformation dans Lucan consiste dans un premier temps à nettoyer les données entrantes pour obtenir des données correctes, complètes, cohérentes et non ambigües[16].

Le nettoyage des données se fait en fonction de certains critères. En effet, il est spécialement requis quand on souhaite intégrer des sources hétérogènes de données, puisqu'il aborde les problèmes de détection de redondance, de suppression des erreurs et des incohérences des données afin d'améliorer la qualité de celles-ci[17].

Généralement, les données biologiques ont un taux problématique assez élevé aux niveaux de leur nomenclature, celui-ci peut être résolu par l'intermédiaire d'une méthode de transformation pour l'analyse future de ces données. Le plus souvent , les différentes étapes sont réalisées telles que[18] :

- ✓ La récupération des données.
- ✓ La standardisation de la nomenclature :

- Aucun espace.
- Utilisation des « _ » afin de séparer des mots.
- Limitation de la longueur des libellées de colonnes (10 caractères maximum).

✓ La surpression des données inintéressantes et des doublons si nécessaires.

✓ La clusterisation ou fusion des données similaires.

✓ Le choix du type de données.

La transformation est l'étape la plus difficile en termes de travail, et la plus coûteuse en termes de temps parmi les trois étapes du processus d'ETL, elle requière parfois des manipulations très précises des données. Elle peut passer d'une simple conversion des données à des techniques de transformation très complexe[19].

e. Système LUCAN et vue d'ensemble des fichiers des plugins

Lucan est la partie back-end de Kibio, il est composé de différents modules, chacun apportant une fonctionnalité au logiciel. Pour le moment, le logiciel étant en phase de production, il était donc possible de mettre en place les plugins, qui rappelons l'était d'ailleurs l'objectif principal de ce stage.

Un plugin correspond à un dossier dans une arborescence dans le dossier « Plugins » de LUCAN. Chaque base de données traitée contient plusieurs fichiers permettant d'utiliser le plugin dans LUCAN.

Ces dossiers doivent être constitués de fichier indispensable au bon fonctionnement d'un plugin :

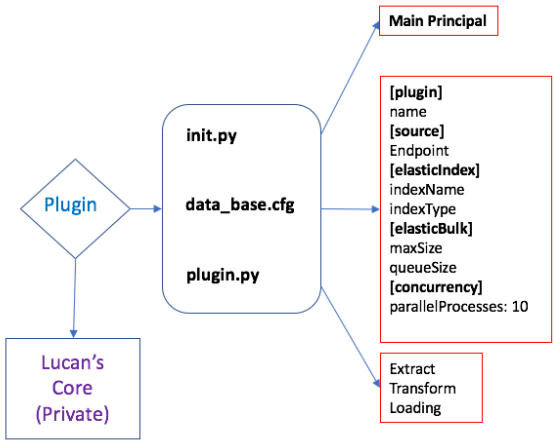


Figure 5 : système d'insertion de plugins d'intégration de base de données

- ➢ __init__.py : c'est le fichier d'initialisation du plugin.
- ➢ Plugin.py : contient le code permettant d'extraire, de transformer et de charger les données dans ElasticSearch. Ce fichier représente la classe principale de notre plugin.
- ➢ data_base.cfg : contient tous les paramètres de configuration d'ElasticSearch permettant une intégration optimale des données dans celui-ci.

2. Chargement des données au cluster de calcul

a. Principe de chargement des données

Afin d'accéder aux données et d'effectuer des opérations de haut niveau fournit par KIBIO, un nouveau système d'organisation des données moléculaires a été développé. Pour cela, il important de charger les données dans un entrepôt, afin de pouvoir y accéder efficacement.

Dans cette partie, nous verrons les outils principalement liés au processus de stockage ainsi que les outils d'accession et de vérification des données.

b. Serveur de calcul

Le laboratoire d'Arnaud DROIT possède à sa disposition deux clusters :

- ✓ Un cluster localisé dans le « CHU Centre et Mère pour Enfant ».
- ✓ Un cluster localisé à « l'université de Victoria » en Colombie-Britannique. Ces serveurs appartiennent à « Calcul Canada » et sont montés sur une architecture partagée.

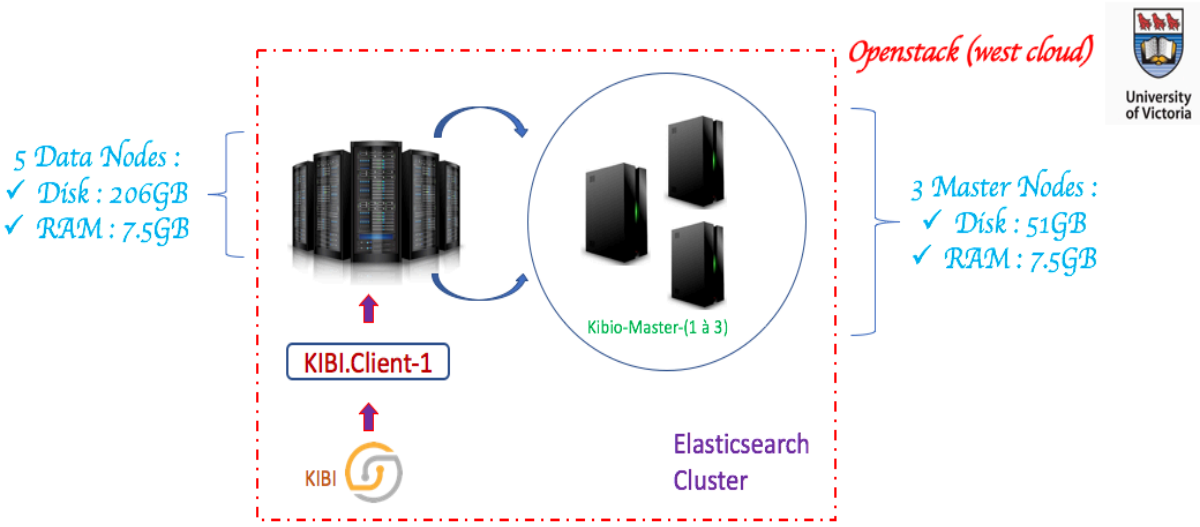


Figure 6 : Architecture du serveur de production

Au cours du stage, le serveur localisé au CHU est utilisé comme serveur de test. Lors du développement de nos plugins informatiques, les données ont été chargées dans un premier temps dans le serveur du CHU, afin de vérifier la bonne standardisation des données.

Après vérification de la bonne fonctionnalité des plugins, elles ont été intégrées dans la version de production de KIBIO sur un dépôt git. Puis dans un second temps, exécutés pour extraire, transformer et charger les données sur le serveur de production situé en Colombie britannique.

Ce serveur de production est constitué de 5 nœuds de données et chacun de ces nœuds est pointé vers des nœuds maître qui gère l'état du cluster. Les nœuds maître ont des actions tel que la création ou la suppression d'un index. Concernant les nœuds de données, ils sont généralement utilisés pour gérer les documents qui ont été indexés. Les opérations courantes de CRUD « Create, Read, Update, Delete », ainsi que la recherche et l'agrégation des données sont leurs tâches principales.

Néanmoins, il faut savoir que les opérations qu'effectuent les nœuds de donnée, nécessitent une grande quantité de RAM et de puissants processeurs, puisque ce sont eux qui vont réaliser les recherches dans les jeux de données massifs, d'où une RAM importante (60GB par nœud), des CPUs octa-core et un espace de stockage de plus de 200Go par nœud.

Il est à noter qu'Elasticsearch a été installé de façon que l'ensemble des serveurs soient reliés entre-eux formant ainsi un cluster de nœuds de recherche.

c. Fonctionnalité d'Elasticsearch

Elasticsearch est considéré comme un outil de stockage, de recherche et d'analyse de données structurées et non structurées.

L'un des plus grands avantages de l'utilisation de cet outil est sa capacité à stocker les données d'une manière astucieuse, ce qui le rend rapide pour la recherche d'informations. En effet les données sont stockées en format JSON « JavaScript Object Notation » qui est un format de structuration de données. Ce format est principalement utilisé pour transmettre des données entre un serveur et une application web. Ainsi, l'insertion du « JSON » dans Elasticsearch, nous permet d'interroger facilement les donner pour les récupérer.

d. Indexation et programmation parallèle dans Elasticsearch

Avant l'insertion des données, nous devons créer un index qui contiendra nos données d'une base de données quelconque. Cependant, il peut arriver qu'un index soit trop lourd en termes d'espace mémoire et donc, ne pourra pas être chargé sur le serveur de calcul.

Pour faire face à cette problématique, Elasticsearch fournit la possibilité de subdiviser notre index en plusieurs pièces appelées « SHARDS » qui

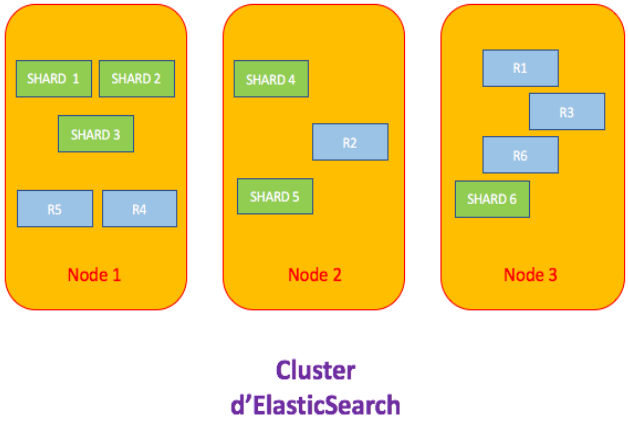


Figure 7 : Diagramme d'un cluster d'ElasticSearch

sont des entités indépendantes et pleinement fonctionnelles. Ainsi lors de la création de notre index, nous pouvons définir le nombre de « SHARDS » que l'on souhaite héberger sur notre cluster.

L'utilisation de ces « SHARDS » est d'une très grande importance, puisqu'elle nous permet de distribuer et d'effectuer des opérations parallèles ce qui augmente la performance et la rapidité de chargement de nos données sur notre cluster de calcul.

Autre spécificité d'Elasticsearch est, d'avoir la possibilité de faire une ou plusieurs copies de nos « SHARDS ». Ces « REPLICATS » fournissant dans un premiers temps une bonne disponibilité de nos données au cas où il y aurait un problème au niveau de SHARDS ou du « NŒUD » en question. Dans un second temps, Elasticsearch aura la possibilité d'effectuer une recherche en parallèle sur tous les réplicas augmentant ainsi la recherche de ces données.

e. Accès à Elasticsearch avec CEREBRO

Elasticsearch est un outil utilisé pour permettre la recherche rapide des données. La seule façon d'en tirer parti des capacités de celui-ci c'est d'utiliser des outils de surveillance d'aide à la supervision notre cluster.

L'outil de supervision des données qui sera donc utilisé est « CEREBRO ». C'est un outil d'administration web construit en utilisant Scala, Play Framework, AngularJS et Bootstrap. Le menu principal de CEREBRO sera composé d'un ensemble vital d'informations montrant l'état de notre cluster d'Elasticsearch.

Afin d'exploiter le potentiel de cet outil, CEREBRO a été préconfiguré avec l'adresse d'Elasticsearch appropriée.

3. Vérification et outils de visualisation des données

a. Objectifs de la visualisation

La visualisation des données est d'une aide précieuse pour permettre aux chercheurs de présenter leurs données dans des graphiques sophistiqués. En effet, en illustrant les relations existantes entre ces données, les connections de celles-ci pourraient nous permettre de nouvelles découvertes scientifiques[20].

Cependant, les méthodes de visualisation sont dépendantes du contexte de requêtage et du domaine scientifique étudié. Elles diffèrent d'un certain nombre d'aspects, tel que les propriétés des données à visualiser, les types d'intéraction et les tâches concrètes à résoudre[21].

Dans le but de visualiser toutes ces propriétés, des outils de visualisation des données biologiques ont été considérablement améliorés durant ces dernières années, mais ils sont encore insuffisant pour certains ensemble de données[22].

b. Outils de visualisation KIBI

Afin de visualiser nos données de médecine moléculaire, nous avons utilisé KIBI, qui est une plateforme gratuite et libre d'accès, il est construit à partir de « KIBANA ».

Figure 8 : visualisation des données présentent dans ElasticSearch avec KIBI

Développer par la société « SIREN SOLUTION », il permet d'effectuer des analyses complexes sur des larges volumes de données en fournissant des visualisations personnalisables par l'intermédiaire d'une interface graphique.

En effet, les visualisations seront réalisées sur les requêtes d'Elasticsearch, puisque les données seront par avance intégrées dans celui-ci. L'autre particularité de KIBI est également de visualiser les interactions des différentes bases de données à partir des identifiants de références communes entrent celles-ci.

4. Module d'interface web

a. Objectifs de l'interface web et ajout de fonctionnalité

De nos jours, beaucoup de logiciel en bioinformatique disposent d'interfaces web. Généralement le logiciel doit être téléchargé, mais celui-ci est souvent associé à un service en ligne. La publication d'un outil à partir d'une application web le rend plus accessible et plus facile à maintenir. Il est donc tout à fait naturel que plusieurs outils en bioinformatique soient dotés d'une interface web.

Un site web a déjà été mis en place par le doctorant REGIS-ONGARO-CARCY. Il permet dans un premier temps, de pouvoir donner aux utilisateurs une bref description générale de KIBIO ainsi qu'un accès à celui-ci « Voir Annexe 1-A ».

Néanmoins, il a été décidé, en concertation avec les membres de l'équipe, d'intégrer dans le site web « KIBIO », un onglet permettant l'insertion d'un « fichier csv » ainsi que les paramètres de configuration qui permettront à LUCAN d'intégrer les données automatiquement dans ElasticSearch.

b. Mise en place de la partie « Front-end » de l'onglet web

La partie front-end correspond à la partie du code qui sera reçu par notre navigateur web. De manière à pouvoir intégrer un nouvel onglet au site déjà mise en place, nous avons utilisé différents langages web tels que :

- HTML (HyperText Markup Language) : Considéré comme un langage de description, il permet de structurer et d'afficher différents objets sur un écran. Il est essentiellement statique, puisqu'il décrit une scène qui sera envoyée vers un client.
- CSS (Cascading Style Sheets) : C'est un Language de feuille de style qui vient compléter le HTML pour gérer l'apparence de la page.
- JavaScript : C'est un language très puissant coté client, le plus souvent utilisé pour ajouter à notre page web un comportement dynamique, stocker de l'information, et manipuler les requêtes et les réponses d'un site internet.

Pour faciliter la programmation de notre site web, deux Framework ont été très utile pour l'amélioration visuelle de notre site, car ils constituent un ensemble d'outils permettant de nous faciliter le travail :

- Bootstrap : Framework développé en HTML, CSS, JavaScript. Il permet de concevoir un site internet rapidement ainsi que la gestion de l'aspect visuel d'un site internet. Il apporte également du style pour les boutons, les formulaires, la navigation.
- JQuery : Framework développé en JavaScript constitué d'un ensemble d'outil développé afin de nous simplifier l'utilisation du JavaScript par l'intermédiaire d'un ensemble d'instruction simple, logique et facile à comprendre.

c. Mise en place de la partie « Back-end de l'onglet web

Après la mise en place de la partie « Front End », il était intéressant de mettre en place le côté Back-end, qui met en évidence tout ce que l'utilisateur ne peut voir sur le navigateur.

De façon à pouvoir insérer les paramètres de configurations ainsi que le fichier csv dans le serveur, nous avons utilisé comme langage de programmation côté serveur « NodeJS » ainsi que son Framework « ExpressJS » qui fournit un ensemble de fonctionnalités web.

En effet, NodeJS est de plus en plus utilisé, puisque c'est un système complètement ou partiellement asynchrone permettant d'accélérer les applications web.

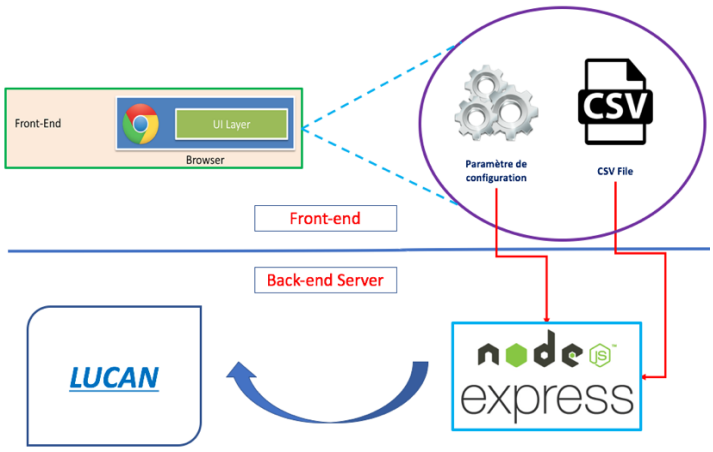


Figure 9: Architecture de la partie Back-end souhaitée

III. Résultats

1. Les bases de données intégrées dans ElasticSearch

Différentes bases de données ont été intégrées dans le cluster d'ElasticSearch par l'intermédiaire de « Lucan » afin de pouvoir extraire les connaissances et de les visualiser graphiquement à partir de « KIBANA ».

Au cours de ce stage, j'ai intégré 20 bases de données « Voir figure 10 ». Lors de la recherche d'informations concernant ces bases de données, une attention toute particulière a été portée sur les relations qu'elles peuvent avoir entrent-elles. Il est évident que plus nous possédons des relations entre nos bases de données plus que nous aurons un ensemble continue d'informations.

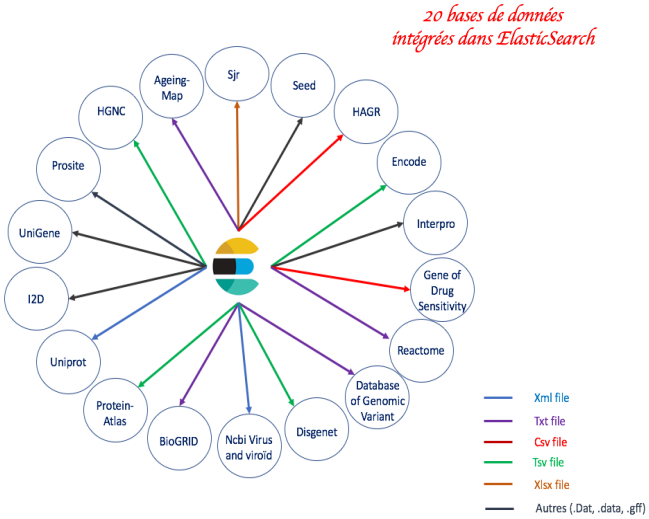


Figure 10 : Les bases de données intégrées dans ElasticSearch

Ainsi, dans un premier temps, une analyse sur les bases de données déjà intégrées dans Kibio a été effectué. Parmi celles déjà présente dans Kibio, une avait particulièrement retenu notre attention. Il s'agit de PubMed, qui est le principal moteur de recherche de données bibliographiques de la biologie et de la médecine. Il nous a donc paru évident de chercher les bases de données qui pourraient non

seulement être reliées à PubMed par l'intermédiaire d'un id de référence, mais également nous apporter des informations supplémentaires.

Ainsi, la première base de données sur laquelle il fût intéressant de travailler était « SJR (Scimago Journal & Country Rank) ». Ce choix était dû au fait qu'il semblerait que 54,2% des articles de SJR sont indexés dans PubMed. En effet, le fait d'avoir une relation existante entre 2 bases, permet d'apporter des informations supplémentaires au données déjà existantes dans PubMed[23].

Cette démarche d'analyse a été suivie tout au long du stage, afin de déterminer les bases de données qui devront être intégrées. Ainsi, 20 bases de données ont été sélectionnées pour KIBIO.

2. Les plugins intégrés dans LUCAN

a. Complexité optimale de nos plugins

Après avoir obtenu la certitude sur la fonctionnalité de mes plugins, une évaluation a été faite sur l'efficacité et la rapidité en mesurant la complexité l'algorithme de transformation des données.

Pour cela, les différentes étapes de l'algorithme ont été passées en revue afin d'évaluer son temps d'exécution. Pour ce faire, un coût en temps à chaque instruction a été attribué et enfin pour finir un comptage du nombre d'exécutions de chacune des instructions a été pratiqué.

L'objectif fixé a été d'optimiser au maximum notre code afin d'améliorer la rapidité de chargement des données dans ElasticSearch quand cela était possible. En effet, un ordre de grandeur du temps nécessaire à l'exécution d'un algorithme de type $O(n^2)$ pour une base de données contenant 1 000 000 documents soit n = 1 000 000, prendrais un temps d'exécution de 2.8 heures.

D'ailleurs, dans le présent cas, la vitesse de chargement des données dépend fortement du processus de transformation. La fonction « yield » de python est utilisé pour le processus de chargement, il permet non seulement d'économiser de la mémoire, mais aussi et surtout de masquer la complexité d'un algorithme derrière une API classique d'itération.

L'optimisation de nos plugins fût une étape très importante afin de réduire le coût d'attente de chargement de nos données dans ElasticSearch.

b. La mise en place des plugins

Les données extraites des bases de données ne sont pas forcément dans l'état dans lequel elles seront stockées. Nous les avons reformatées, nettoyées afin d'éliminer les valeurs

incohérentes ainsi que les doublons. Cette étape nous a en effet permis d'assurer la validité des données.

La construction du plugin nécessite de disposer d'une source de données. Chaque plugin développé dépendra de la base de données sur laquelle nous avons travaillé.

Exemple de transformation d'une base de données « structurée » : « SJR (Scimago Journal & Country Rank » :

SJR est un portail web public qui inclut les revues ainsi que les mesures d'influences statistiques de revues académiques scientifique. Il prend en compte le nombre de fois que les citations sont apparues dans des revues et l'importance des journaux d'où proviennent ces citations.

Le format de cette base de données est de type « Xlsx », et les données sont rangées par date, soit de l'année 1999 à 2016. Les données sont structurées de la manière suivante, les entêtes des colonnes en première ligne et les colonnes réelles restent toujours dans cette position.

Ainsi avant de traiter les données, il convient de télécharger tous les fichiers Xlsx de manière automatique, provenant des années précédentes jusqu'à aujourd'hui. Pour ce faire, le lien web sera récupéré, il permettra d'accéder aux fichiers de la source concerné.

En prenant l'exemple de la base de données pour l'année 2016 : « *http://www.scimagojr.com/journalrank.php?out=xls&year=**2016*** ». Ainsi en utilisant la fonction range on peut accéder aux liens permettant d'accéder aux fichiers des bases de données.

```
sjr_files = []
for year in range(1999, 2017):
    total_link = "{}{}".format(self.endpoint, year)
    # Download file from specific link
    sjr_files.append({'year':year, 'data':self.download(total_link, "{}.xlsx".format(year))})
```

Figure 11 : Code Python3.5 montrant la fonction range permettant d'accéder aux fichiers de la base de données "SJR" des années 1999 à 2016

A chaque tour de boucle, la variable « total_link » contiendra le lien du téléchargement du fichier ayant pour date la variable « year ». La variable « self.endpoint » contient le lien *http://www.scimagojr.com/journalrank.php?out=xls&year=*. Tous les liens du téléchargement seront stockés dans la liste « srj_files » de façon à être stocké efficacement et réutilisable par la suite.

Après avoir récupéré les fichiers des années correspondantes, l'étape de nettoyage et de transformation a pu être mise en place. Ainsi, afin d'extraire les données d'un fichier Microsoft Excel, nous avons utilisé la librairie python « xlrd ».

Un aperçu de transformation le plus souvent rencontré est l'apparition de plusieurs valeurs associées à un « header ». Par exemple, le journal « A Cancer Journal for Clinicians » contient un Issn de « ISSN 15424863, 00079235 ». Dans ce cas précis, une suppression de ISSN et une séparation en prenant en compte comme délimiteur la «, » a été effectué, afin d'obtenir une liste [15424863, 00079235].

A la suite de l'étape de transformation, les données ont été rangées dans une structure adaptée à ElasticSearch. En effet, ce moteur de recherche s'appuie sur une base documentaire NoSQL interne et utilise le format JSON pour le stockage des documents. De façon à permettre l'indexation des documents, après chaque lecture de ligne du fichier, les données sont arrangées sous forme de dictionnaire, afin d'être envoyé à ElasticSearch.

La vérification de la conformité des données s'est faite avec CEREBIO pour la détection des erreurs au cours de l'étape de chargement des données. En cas d'erreur une correction a été effectuée dans l'immédiat.

Exemple de transformation d'une base de données « semi-structurée » : « Uniprot » :

Uniprot est une base de données de séquence protéique issue de la consolidation de l'ensemble des données produites par la communauté scientifique. Au sein de cette dernière, chaque séquence est accompagnée d'un ensemble de métadonnées et de liens vers de nombreuses autres base de données.

En ce qui concerne les entrées d'Uniprot, elles sont disponibles en format XML, d'où l'appellation « base de données semi-structurées ». La réelle difficulté est apparue lors de l'extraction et la transformation des données puisque l'accès aux informations et la transformation de celles-ci furent assez compliqués.

Après de nombreuses recherches sur le web, il fût intéressant de constater la présence d'une librairie « Biopython » permettant de parser efficacement les données d'Uniprot, il s'agit du module SeqIO du package Biopython. Ici la fonction BIO.SeqIO.parse() a été utilisée, celui-ci prend en entrée le fichier Uniprot à manipuler et renvoie un Ittérateur donnant des objets SeqRecord.

A partir des objets SeqRecord, nous avons pu récupérer les informations de chaque entrée d'Uniprot de manière à les charger dans ElasticSearch.

3. Visualisation des données intégrées sur KIBANA

a. Observation des données intégrées dans Uniprot

L'intérêt de ce processus d'intégration de base de données dans ElasticSearch, est la possibilité de les visualiser par l'intermédiaire de KIBI. Prenons l'exemple d'Uniprot, qui est la base de données protéiques de référence.

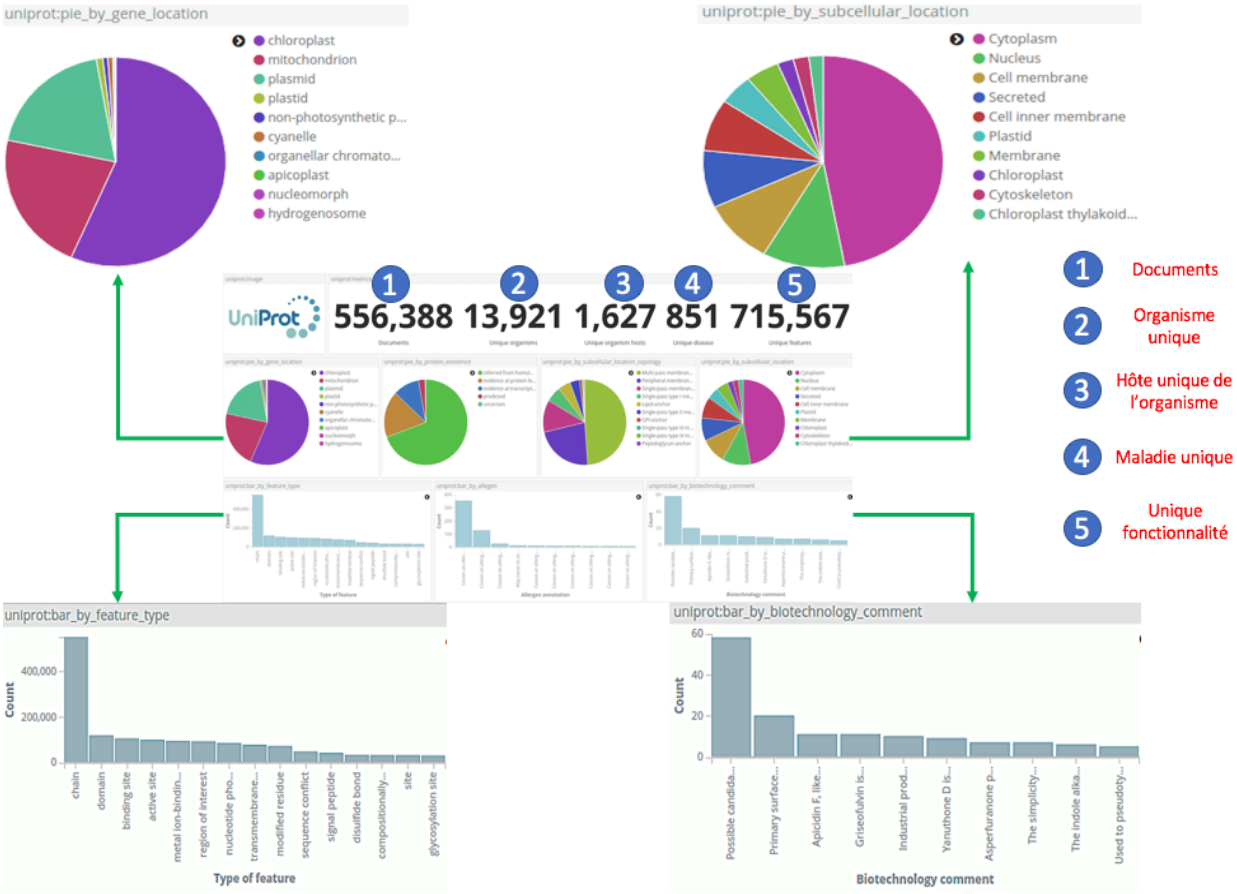


Figure 12 : visualisation des données d'Uniprot avec KIBI

Plusieurs types de visualisation ont été réalisés en fonction des requêtes fournies. Parmi ces multitudes de « charts » proposées par « KIBI », deux nous a semblé être plus appropriés par rapport aux types de données d'Uniprot « voir figure », il s'agit du camembert et du graphe à bar vertical.

On peut donc construire des visualisations à partir d'une métrique d'agrégation d'ElasticSearch, qui va simplement agréger les données en fonction des valeurs extraites de part et d'autre du document.

Une des options courantes existante pour nos visualisations est l'option « split slice », qui montre de quelle manière ces agrégations du camembert doivent être affichées. Ainsi, de cette manière, on va générer une tranche pour chaque compartiment de notre camembert.

En sélectionnant une agrégation de terme tel que « by_gene_location » pour le camembert A et « by_subcellular_location » pour le camembert B, ElasticSearch va effectuer une agrégation métrique qui va déterminer les différentes tranches de nos camemberts en prenant en compte le nombre total de document dans Uniprot.

On retrouve dans nos 2 camemberts une tranche par seau de localisation spécifique qui dépend du nombre de fois qu'on le retrouve dans les documents. Pour celui correspondant à la localisation des gènes, on observe que parmi les 556,388 documents présents dans « Uniprot », une grande majorité des gènes semble être localisée dans le chloroplaste, contrairement aux protéines qui sont localisées majoritairement dans cytoplasme.

Autre mode de visualisation proposé par « KIBANA », c'est la visualisation de la barre verticale. Dans ce cas de visualisation, chaque valeur sur notre axe de x obtiendra ses propres barres, et il n'y aura pas d'interpolation entre les valeurs de ces barres. Il placera les barres pour chaque valeur de l'axe des x à côté de l'autre.

Nous avons ainsi réalisé, un graphe a bar vertical en fonction des annotations de séquence. Ces annotations de séquences permettent de décrire des régions ou des sites d'intérêt dans les séquences protéiques. A partir de ce graphique, on peut observer que la grande majorité des protéines présentes dans Uniprot sont des extensions de chaine polypeptidique de protéine mature. On retrouve également dans Uniprot des :

- ✓ Domaines protéiques
- ✓ Sites de liaisons pour des groupes chimique qui correspondent généralement à des coenzymes, des groupes prosthétiques.
- ✓ Sites actives qui correspondent aux acides animés directement impliqué dans l'activité d'une enzyme.

Kibana nous a permis également de compter des éléments uniques. Pour cela, nous avons choisi les champs « organisme unique », « Hôte unique de l'organisme », « maladie unique », et « unique fonctionnalité ». Ceci nous a permis de compter combien de valeurs différentes existent dans le document pour ce compartiment.

Si nous effectuons une agrégation de nombre unique sur le champ :

- ➢ Organisme : on observe 13921 organismes différents
- ➢ Hôte de l'organisme : 1627 hôtes pour chaque organisme
- ➢ Maladie unique : 827 maladies unique

b. Analyse de précision : cas de PubMed

Afin de montrer la particularité d'analyse de précision de Kibana, nous utiliserons PubMed du fait de son accessibilité, et de sa reconnaissance de toutes les disciplines scientifiques confondues. En effet, PubMed est le principal moteur de recherche de données bibliographique de l'ensemble de domaines de spécialisation de la biologie de la médecine.

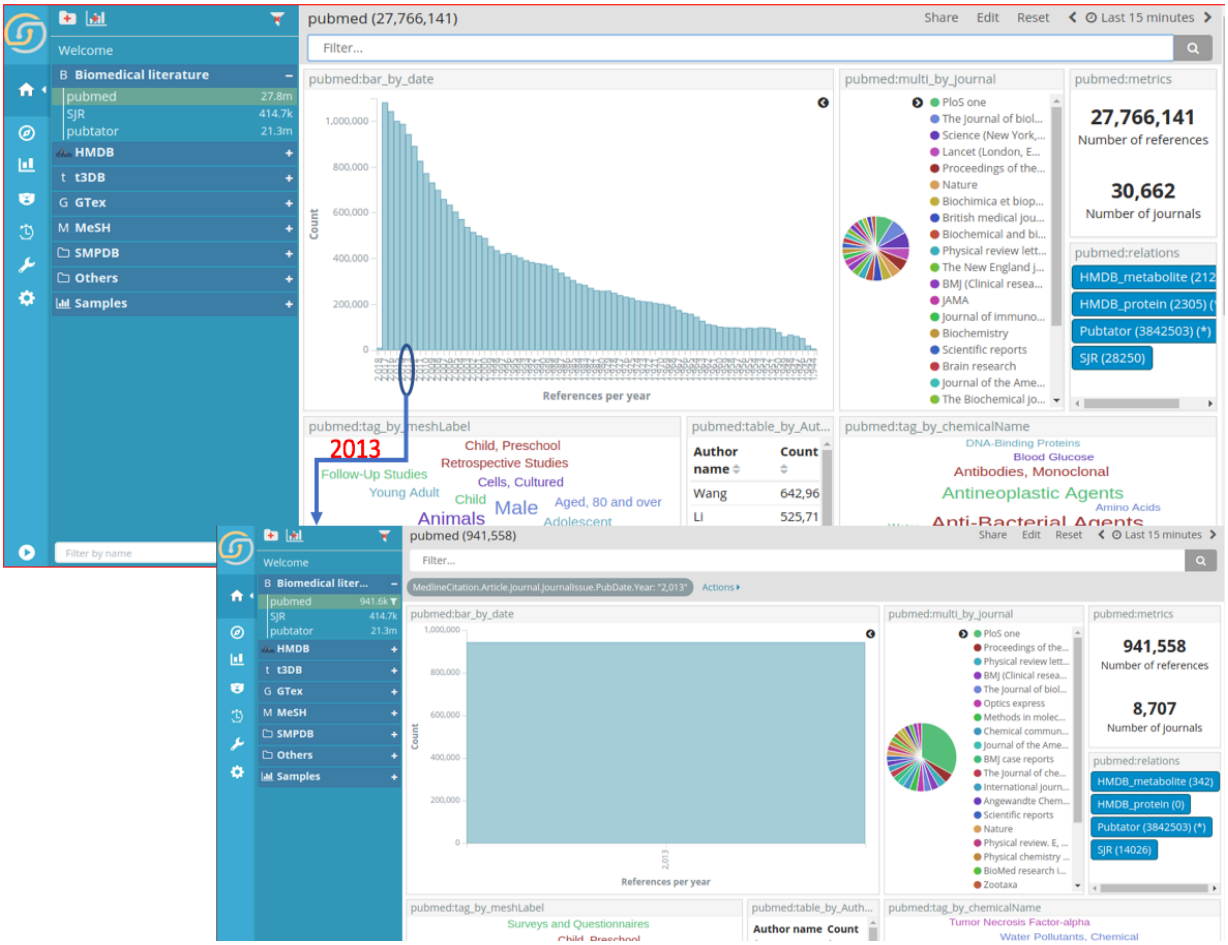


Figure 13 : Visualisation et analyse des données de PubMed à partir de KIBI

Les données de PubMed étant déjà intégrées dans ElasticSearch, des visualisations à partir de KIBANA ont été réalisées afin de visualiser le contenu de celui-ci graphiquement « Voir figure 12 ».

Nous observons sur cette « figure 12 » un graphique à barre vertical démontrant une croissance exponentielle de publication au cours des années. En effet, cette dernière décennie a été marqué par les progrès récents de la recherche à haut débit et de la croissance rapide de la recherche sur les données biologiques à grande échelle.

Ces avancées technologiques telle que l'apparition de séquenceur de nouvelle génération à coïncider à cette croissance exponentielle de la littérature biomédicale. D'ailleurs

on peut observer sur ce graphe, que c'est entre les années 2007 et 2008 qu'on a pu observer une croissante plus forte, date à laquelle est apparue les séquenceurs de nouvelle génération.

L'une des options qu'offre KIBANA est la possibilité d'observer ces données avec précisions. En effet, en sélectionnant l'année 2013 en cliquant sur la barre verticale de l'année correspondante sur l'histogramme, il nous a été possible d'observer plus en détail les données acquises au cours de cette année.

L'observation que nous pouvons effectuer, est qu'en 2013, 941 558 références d'articles et un total de 32 662 journaux ont été publiés sur PubMed. Parmi ces références, une grande majorité d'entre elles proviennent de revues scientifiques tel que :

- Plos One
- The Journal of Biological Chemistry
- Science
- Lancet
- Nature

L'une des grandes particularités de KIBI, c'est la possibilité de détecter les relations entre les bases de données intégrées dans ElasticSearch. De ce fait, il est donc possible d'observer que PubMed est relié aux bases de données « HMDB_metabolite », « HMDB_protein », « Pubtator », « Sjr ».

c. Relation entre toutes les bases de données

Le plugin de haute performance « Siren Join » implémenté dans KIBI permet la recherche relationnelle au sein d'ElasticSearch afin de définir des relations entre les modèles d'index. Ces relations vont former ce qu'on appelle un « graphique des modèles d'index ».

Figure 14 : visualisation des relations entre PubMed et Pubtator

Ainsi en formant un graphique relationnel ente l'index PubMed et Pubtator, on obtient ce graphique montrant toutes les connexions des bases de données dont elles possèdent.

d. Mise en place de l'onglet web

Afin de permettre un accès de Kibio aux utilisateurs, un site web a été mise en place. Cette plateforme contient différents onglets d'informations nécessaires à la compréhension du logiciel. Cependant, il nous a paru intéressant de rajouter un onglet web dans le but d'intégrer automatiquement une base de données uniquement pour le moment au format de fichier « csv ».

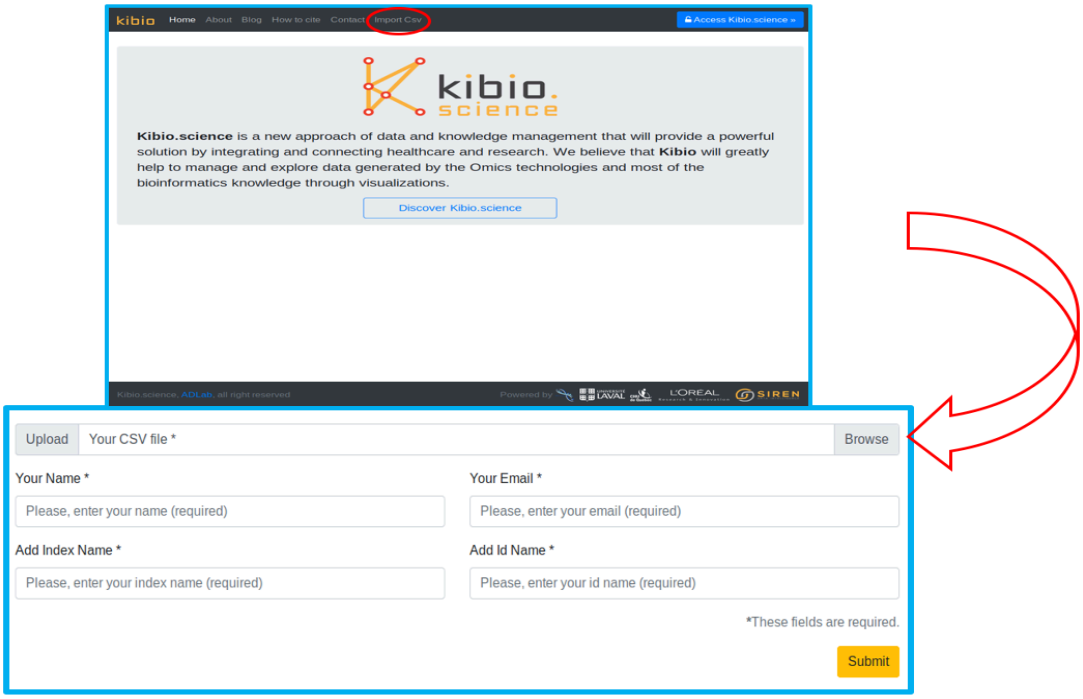


Figure 15 : Insertion d'un formulaire web dans l'onglet "import Csv"

Ainsi un onglet web « Import CSV », contenant un formulaire a été implémenté. Celui-ci permet d'intégrer dans un premier temps, les informations concernant l'utilisateur, et dans un second temps les éléments essentiels et nécessaires pour l'intégration de la base de données d'intérêt dans ElasticSearch.

En effet, on retrouve l'onglet « Index Name » qui correspond au nom que portera la base de données dans ElasticSearch et « Add Id Name » qui correspondra à l'identifiant unique dans la base de données. Néanmoins, de part cette interface, il a été nécessaire de mettre en œuvre la partie back-end ayant pour rôle d'intégrer ces données dans le serveur.

Avec Node.JS, nous avons pu gérer les données du client à l'aide de la méthode POST qui est une méthode de transfert des données de formulaires au serveur également appelé

« construction de l'API REST Node.js Express ». Les données du formulaire ont donc été récupérées et transférées au serveur, dans l'attente d'être utilisé par LUCAN.

IV. Discussion

Kibio.science est une plateforme basée sur le web permettant d'analyser et de visualiser les données de la médecine moléculaire. Dans ce rapport, nous présentons plus particulièrement « LUCAN » qui est la partie Back-end de KIBIO, qui s'organise autour de la gestion des données. En utilisant le processus d'ETL, LUCAN permet d'extraire, transformer et de charger les données dans ElasticSearch, à partir de différents plugins intégrés dans celui-ci. Une optimisation de la complexité algorithmique des plugins a été prise en compte de façon à permettre un chargement rapide des données dans le cluster de calcul ElasticSearch.

Le portail Kibio.science permet d'accéder à la Platform KIBI de façon à visualiser et analyser le contenu des bases de données intégrées par LUCAN. Ainsi, afin de démontrer l'intérêt de l'intégration des données, nous avons présenté trois différentes bases de données telles que Uniprot, PubMed, Sjr. Dans l'exemple d'Uniprot, il fût intéressant d'observer que les gènes sont majoritairement retrouvés dans le chloroplaste, les mitochondries, et les plasmides tandis que les protéines sont majoritairement présentes dans le cytoplasme, noyau et membrane cellulaire.

Visualisation

=➔ Les gènes sont majoritairement localisé dans le chloroplaste, mitochondrie, plasmide ;

➔ Les protéines sont majoritairement présentes dans cytoplasme, noyau, membrane cellulaire.

➔ Peut-être que les protéine d'Uniprot provienne généralement de chez les végétaux ;

➔Croissance de donnée de littérature en particulièrement depuis l'apparition de séquenceur de nouvelle génération.

➔ Appel Lucan permettant de récupérer le ficher csv ainsi que les paramètres de configuration afin de pouvoir traiter les données.

➔ Application machine Learning

(Amélioration envisagée)

V. Conclusion

VI. Bibliographie

1. Hood, L. & Rowen, L. The human genome project: big science transforms biology and medicine. *Genome Med.* **5,** 79 (2013).
2. Cook, C. E. *et al.* The European Bioinformatics Institute in 2016: Data growth and integration. *Nucleic Acids Res.* **44,** D20–D26 (2016).
3. Shaer, O., Nov, O., Westendorf, L. & Ball, M. Communicating Personal Genomic Information to Non-experts: A New Frontier for Human-Computer Interaction. *Found. Trends® Human–Computer Interact.* **11,** 1–62 (2017).
4. Sboner, A., Mu, X. J., Greenbaum, D., Auerbach, R. K. & Gerstein, M. B. The real cost of sequencing: higher than you think! *Genome Biol.* **12,** 125 (2011).
5. Chaussabel, D. & Pulendran, B. A vision and a prescription for big data–enabled medicine. *Nat. Immunol.* **16,** 435 (2015).
6. Rigden, D. J. & Fernández, X. M. The 2018 Nucleic Acids Research database issue and the online molecular biology database collection. *Nucleic Acids Res.* **46,** D1–D7 (2018).
7. Helmy, M., Crits-Christoph, A. & Bader, G. D. Ten Simple Rules for Developing Public Biological Databases. *PLOS Comput. Biol.* **12,** e1005128 (2016).
8. Dai, L., Gao, X., Guo, Y., Xiao, J. & Zhang, Z. Bioinformatics clouds for big data manipulation. *Biol. Direct* **7,** 43 (2012).
9. Shah, S. P., Huang, Y., Xu, T. & Ouellette, F. Atlas – a data warehouse for integrative bioinformatics. *BMC Bioinformatics* 16 (2005).
10. Lee, T. J. *et al.* BioWarehouse: a bioinformatics database warehouse toolkit. *BMC Bioinformatics* 14 (2006).
11. Inmon, W. H. Building the Data Warehouse. 428
12. Gosain, A. & Arora, A. Security Issues in Data Warehouse: A Systematic Review. *Procedia Comput. Sci.* **48,** 149–157 (2015).
13. Astriani, W. & Trisminingsih, R. Extraction, Transformation, and Loading (ETL) Module for Hotspot Spatial Data Warehouse Using Geokettle. *Procedia Environ. Sci.* **33,** 626–634 (2016).
14. Merelli, I., Pérez-Sánchez, H., Gesing, S. & D'Agostino, D. Managing, Analysing, and Integrating Big Data in Medical Bioinformatics: Open Problems and Future Perspectives. *BioMed Res. Int.* **2014,** 1–13 (2014).

15. El-Sappagh, S. H. A., Hendawi, A. M. A. & El Bastawissy, A. H. A proposed model for data warehouse ETL processes. *J. King Saud Univ. - Comput. Inf. Sci.* **23,** 91–104 (2011).
16. Agarwal, A. K. & Badal, N. A novel approach for intelligent distribution of data warehouses. *Egypt. Inform. J.* **17,** 147–159 (2016).
17. Lenzerini, M. Data Integration: A Theoretical Perspective. 14
18. Herbert, K. G. & Wang, J. T. L. Biological data cleaning: a case study. *Int. J. Inf. Qual.* **1,** 60 (2007).
19. Denney, M. J., Long, D. M., Armistead, M. G., Anderson, J. L. & Conway, B. N. Validating the extract, transform, load process used to populate a large clinical research database. *Int. J. Med. Inf.* **94,** 271–274 (2016).
20. Habermann, B., Villaveces, J. & Koti, P. Tools for visualization and analysis of molecular networks, pathways, and -omics data. *Adv. Appl. Bioinforma. Chem.* 11 (2015). doi:10.2147/AABC.S63534
21. Kerren, A., Kucher, K., Li, Y.-F. & Schreiber, F. BioVis Explorer: A visual guide for biological data visualization techniques. *PLOS ONE* **12,** e0187341 (2017).
22. O'Donoghue, S. I. *et al.* Visualizing biological data—now and in the future. *Nat. Methods* **7,** S2–S4 (2010).
23. Jamali, J., SalehiMarzijarani, M. & Ayatollahi, S. Factors Affecting Journal Quality Indicator in Scopus (SCImago Journal Rank) in Obstetrics and Gynecology Journals: a Longitudinal Study (1999-2013). *Acta Inform. Medica* **22,** 385 (2014).