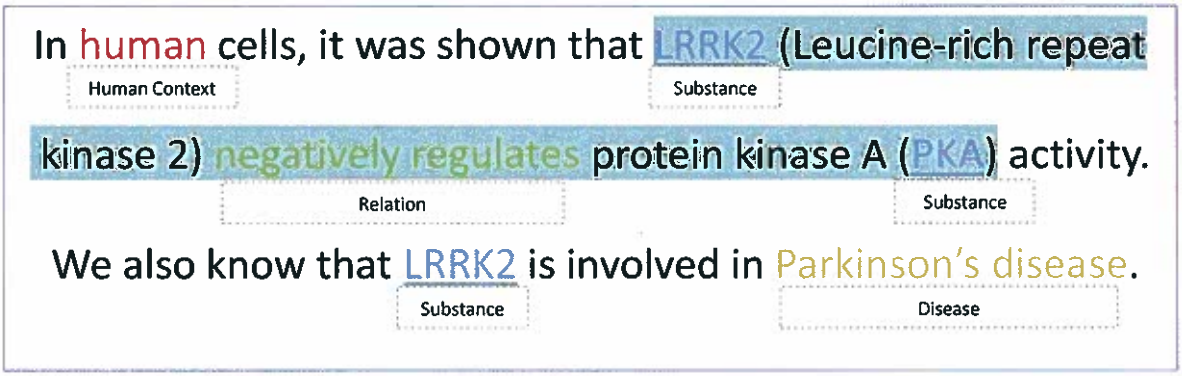
université
PARIS-SACLAY

DASSAULT SYSTEMES

RAPPORT DE STAGE
Du « Text-Mining » pour la biologie des systèmes

MASTER AMI2B



YAHYAOUI Ines (INTERN)
8/12/2016

Note de confidentialité

Ce rapport de stage de fin de MASTER contient des informations confidentielles de DASSAULT SYSTEMES. Toute copie ou divulgation de son contenu est strictement interdite.

Cette copie doit impérativement être rendue à la fin de la soutenance.

Remerciements

Je tiens à remercier, tout premièrement, la personne sans qui je n'aurai pu passer ces six derniers mois, mon maître de stage, Richard MARQUIS. Son accueil, son soutien et ses encouragements ont été mes meilleurs atouts et n'ont fait qu'accroitre ma motivation. Il a toujours été à l'écoute et m'a tout de suite intégré dans son équipe en faisant en sorte que mon travail soit collaboratif, comme je le souhaitais.

Je remercie également toutes les autres personnes avec qui j'ai collaboré, entre autre et surtout Fréderic POTIER, mon coéquipier. Il a fait en sorte que je ne me repose jamais sur mes acquis et que je me surpasse de jour en jour. Il m'a transmis une partie de son savoir, et a tout fait pour que mon stage me soit le plus bénéfique.

Une autre personne a aussi participé à ma montée en compétence sur une nouvelle technologie, un ancien « BIBS », qui n'était pas dans mon équipe ni même dans mon département, mais que je n'ai pas hésité à embêter : Damien DONON. Pour tout le temps qu'il m'a accordé, je le remercie.

Je n'oublie pas tous les autres collaborateurs avec qui j'ai pu échanger, qui m'ont aidé à avancer, et tous mes collègues qui ont fait de ce stage une belle aventure.

Je remercie également mes professeurs, qui m'ont donnée les bases pour accomplir ce projet et j'espère encore beaucoup d'autres. A Christine FROIDEVAUX, Olivier LESPINET et tous les autres transmetteurs de savoir du master AMI2B, merci.

Résumé

Ce rapport de stage est la finalité de ma formation, un Master 2 en Bio-informatique, le Master AMI2B de l'université Paris Saclay.

J'y explique le contexte dans lequel j'ai passé mes six mois de stage, et le sujet que j'ai traité : le text-mining appliqué à la biologie des systèmes. Pour ce travail de recherche, j'ai utilisé une technologie nommée CloudView et créée par EXALEAD, une des filiales de la société dans laquelle j'ai effectué mon stage. Cette technologie a été une pièce maitresse de mon étude, et un des objectifs était de montrer que les capacités de cet outil offraient de nombreux avantages à une telle étude.

Ceci est donc une explication détaillée du processus qui permet de passer de textes scientifiques à des graphes d'interactions entre protéines. Ce processus est le fruit de ma réflexion sur le sujet, réflexion qui m'a permis de monter en compétences dans de nombreux domaines. Ce stage a donc été une suite logique à ma formation, mais n'est absolument pas une finalité à ma poursuite du savoir.

SOMMAIRE

Introduction

Dans le cadre de mon stage de fin de Master, j'ai été amenée à collaborer au sein d'une des équipes de la société DASSAULT SYSTEMES, plus précisément de la marque BIOVIA.

Durant ce stage de 6 mois, j'ai pu participer, réfléchir, comprendre et apporter des solutions à des problématiques métiers dans le domaine de l'industrie du médicament. Dès le début du stage, j'ai fait savoir à mon encadrant, Richard MARQUIS, que je tenais à fournir un travail utile, et avec une réelle valeur ajoutée aux objectifs de l'équipe. Le but est de trouver des solutions innovantes pouvant être intégrées aux futures applications BIOVIA. C'est donc dans ce contexte que je vais détailler ce parcours, les différents objectifs fixés, atteints ou non, les méthodes utilisées et les résultats qui seront discutés.

L'objectif et l'intitulé du stage étaient la recherche et l'implémentation d'une méthode d'inférence de réseaux biologiques, plus particulièrement d'interactions entre protéines. Le but était de pouvoir générer des graphes d'interactions entre protéines, dans un contexte défini – au sein de l'espèce humaine – avec un score de fiabilité correct.

Après un état de l'art et une réflexion sur l'existant et le possible, c'est la piste du text-mining sur des articles qui a été favorisée. En effet, nous nous sommes rendu compte que la meilleure « matière première » à exploiter en termes d'inférence de réseaux biologiques était le corpus de publications scientifiques.

Le text-mining n'ayant jamais été introduit dans des produits du projet Bio Intelligence (à l'origine de BIOVIA), ce travail a beaucoup intéressé, et j'ai donc été amené à rédiger des présentations pour démontrer la faisabilité du projet.

Ce rapport présentera d'abord le contexte et l'environnement dans lequel j'ai passé ces 6 derniers mois, puis je détaillerai le sujet et les objectifs de stage, ainsi que l'état de l'art fait sur le sujet, pour ensuite parler des ressources et des technologies utilisées. Nous verrons ensuite quelles ont été les méthodes implémentées pour parvenir à nos objectifs, pour finir sur une présentation et une discussion des résultats obtenus.

I. Contexte du stage

1. DASSAULT SYSTEMES

Figure 1. Logo de la société DASSAULT SYSTEMES

Le groupe DASSAULT est l'ensemble de trois filiales indépendantes : DASSAULT AVIATION, DASSAULT SYSTEMES et le groupe FIGARO. DASSAULT SYSTEMES est une société d'édition de logiciels spécialisée dans la CAO (Conception Assistée par Ordinateur), avec leur produit phare : CATIA (Conception Assistée Tridimensionnelle Interactive Appliquée).

En 1977, une équipe d'une vingtaine d'employés de DASSAULT AVIATION développe le logiciel CATI qui permet de gérer le processus de conception aéronautique 3D. En 1981, la société DASSAULT SYSTEMES est fondée pour assurer le développement et l'expansion du logiciel qui est alors renommée CATIA.

L'expansion étant allée au-delà de cet unique logiciel, la société DASSAULT SYSTEMES comprend à présent 11 marques dans différents secteurs industriels variés, allant de la finance jusqu'à la biologie, en passant par les secteurs de la géologie ou encore de l'énergie. Société de passion au cœur de l'innovation, elle compte actuellement plus de 15000 collaborateurs, dans près de 40 pays.

Figure 2. Liste des marques de DASSAULT SYSTEMES

2. *BIOVIA R&D France*

En 2014, DASSAULT SYSTEMES rachète Accelrys, et lance alors sa marque dédiée aux sciences de la vie : BIOVIA.

BIOVIA apporte des solutions visant à aider les entreprises concernées à innover dans les domaines biologiques, chimiques et matérielles. Ces solutions tournent toutes autour de la digitalisation du vivant dans le but de pouvoir le simuler.

Les équipes BIOVIA R&D sont nombreuses, chacune travaillant sur un produit différent. Les équipes avec lesquelles j'ai pu collaborer sont celles de Bio Content Manager (BCM), Bio Knowledge Explorer (BEX) et Competitive Intelligence (CI).

BCM est une application en cours de développement de mise à disposition de contenu scientifique qui permettra d'alimenter les autres applications BIOVIA en données chimiques et biologiques connues, mais qui autorisera également l'utilisateur à y intégrer ses propres données.

BEX est une application permettant de générer et visualiser des graphes d'interactions entre protéines, mais également des voies de signalisation en navigant sur les relations entre entités intervenant dans les réseaux biologiques.

CI est un projet, encore dans la première étape de « define », qui permettra de sécuriser le choix d'une cible thérapeutique d'intérêt vis-à-vis des compétiteurs, en navigant dans la connaissance (brevets, publications, essais cliniques, composés chimiques).

C'est en partie pour ce dernier projet que mon travail a servi de preuve de faisabilité en ce qui concerne l'extraction de connaissances dans des données textuelles.

Les équipes de BCM et de CI sont en fait 2 sous-groupes d'une même et grande équipe, et c'est dans ce contexte que j'ai collaboré avec celles-ci.

3. *L'équipe*

MARQUIS Richard

Mon encadrant et le manager de l'équipe travaillant sur le projet CI. Durant mes 6 mois de stage, je l'ai vu travailler à la fois sur ce nouveau projet, afin de montrer son potentiel et sa faisabilité à l'extérieur, mais également aider l'équipe BCM à respecter leur objectifs et leurs délais en leur apportant les compétences manquantes.

POTIER Fréderic

Développeur au sein de l'équipe de Richard, il a pour rôle de définir les finalités des applications –le define – en l'occurrence de BCM et CI. Il a également été l'initiateur du projet de text-mining sur des publications scientifiques, et j'ai donc beaucoup collaboré avec lui.

CHANDRESIS Gabriel

Développeur au sein de l'équipe de Richard, il a un profil technique lui permettant de programmer rapidement pour tout type de besoin.

4. *Organisation de l'équipe*

L'équipe s'organise en itérations de 2 semaines, appelées des « sprints », de sorte que toutes les 2 semaines de nouveaux objectifs soient fixés, et qu'on puisse en faire le bilan à la fin de l'itération. Afin de visualiser l'évolution de notre travail, l'équipe organise des petits points journaliers, d'une quinzaine de minutes, afin que tout le monde échange sur son travail et sur le temps passé sur chaque tâche. J'ai participé à chaque itération et réunion journalière et cela m'a beaucoup servi pour organiser mon travail. Cette méthode d'organisation n'est pas nouvelle, elle porte le nom de « Scrum » (*https://fr.wikipedia.org/wiki/Scrum*) et est souvent utilisée pour réaliser de gros projets.

Note : Le « nous » de ce rapport correspondra à mon équipe et moi-même, plus particulièrement Frédéric POTIER et moi.

II. Objectifs

1. Sujet

Les réseaux biologiques sont un des plus grands secteurs de recherche car ils sont à la base de la biologie des systèmes et donc, absolument primordiaux pour comprendre les mécanismes liés aux maladies et permettre la recherche de nouveaux médicaments. Nous avons choisis de nous concentrer premièrement sur les interactions entre protéines afin de construire des réseaux. Le but était de trouver un moyen d'inférer ces interactions.

La source de données en entrée n'a pas été fixée initialement, et l'un des premiers objectifs était de se positionner sur ce choix.

Une des conditions à respecter était également de pouvoir obtenir des résultats qualifiables en termes de confiance, afin de donner à l'utilisateur une idée de la pertinence de chaque information.

2. Résultats attendus

Les résultats attendus pour ce travail étaient de pouvoir générer, pour une protéine donnée, ou une maladie donnée, le réseau d'interactions de protéines associées. Pour chaque graphe généré, nous avons voulu savoir si les interactions trouvées sont déjà existantes dans une base de données de référence et d'avoir un score de fiabilité pour chacune d'entre elles.

Il était nécessaire de restreindre notre étude à un contexte donné que nous avons défini par le choix d'une espèce, car les protéines peuvent être les même et réagir différemment entre les espèces: La protéine A peut activer la protéine B chez l'homme mais l'inhiber chez la drosophile. Nous avons donc choisi d'étudier les interactions entre protéines uniquement chez les humains. Ce choix est également dû au fait que notre travail tend à s'adresser à l'industrie du médicament.

Ce travail est donc un travail de recherche qui aura pour but de démontrer la faisabilité d'une telle étude, et ainsi de montrer si cette approche est une vraie opportunité pour le produit CI. Dans ce même contexte, le but était également de voir si la technologie utilisée (CloudView par EXALEAD qui sera présentée ci-après) fournissait d'assez bonnes capacités pour pouvoir réaliser ce projet.

III. Etat de l'art

Mon premier mois de stage m'a permis de réaliser un état de l'art assez complet sur le sujet et les objectifs posés. L'inférence d'interactions entre protéines et de réseaux a déjà fait plusieurs fois l'objet de stages et de thèses, et je me suis donc basée sur les états de l'art existants et fournis par mes prédécesseurs. Cela m'a servie de premières pistes à explorer.

Mes premières recherches se sont portées sur les publications scientifiques. Je me suis constituée une sélection d'articles pertinents, relatifs à l'inférence d'interaction entre protéines et aux réseaux [1-7]. J'ai également trouvé des thèses de personnes ayant étudié le sujet pendant des années [27,28]. Du coté des applications existantes, peu sont disponibles gratuitement, mais certaines fournissent des informations sur leurs résultats.

1. Thèses

La première thèse que j'ai étudiée est celle de Pauline GLOAGUEN : « Inférence automatique de modèles de voies de signalisation à partir des données expérimentales ». J'ai pu y trouver un état de l'art assez complet concernant l'inférence et la construction de réseaux biologiques. L'auteur y explique les différentes méthodes d'inférences, notamment l'approche manuelle en faisant de la curation dans la littérature scientifique. Elle parle également d'approche semi-automatique, comme celles basées sur le text-mining, et elle cite plus particulièrement un outil nommé MedScan [19]. Cette application fait de la fouille de données dans les extraits des publications scientifiques. Elle permet donc de reconstruire des réseaux uniquement en étudiant des publications. La dernière approche que l'auteur cite est l'approche automatique. Elle décompose alors ces méthodes en 4 catégories : Modèle statistique, Modèle logique, Systèmes d'équations différentielles et Méthodes à base de règles. L'état de l'art fait pour cette thèse m'a donc permis d'avoir une vision un peu plus dégagée sur les méthodes d'inférences.

La deuxième thèse qui m'a également beaucoup inspirée est celle d'Ithipol SURIYAWONGKUL : « The identification of Target Proteins from Patents ». Il s'agit d'une étude de brevet dans le but d'y extraire des relations protéine-cible. Cette étude se fait à l'aide du text-mining et l'auteur y explique comment elle a créé son dictionnaire de protéines, avec tous les synonymes. Il détaille les différentes phases de tri dans les brevets : déterminer ce qui est une protéine et ce qui n'en est pas, puis déterminer si une protéine est une cible ou non en fonction du contexte. Il émet des hypothèses, sur lesquelles sont fondées toute son étude, entre autres qu'une protéine cible sera uniformément mentionnée dans le texte ou plus fréquemment mentionnée.

Les deux thèses que j'ai étudiées mentionnent le text-mining, l'une le présente comme une des méthodes possibles et l'autre en a fait un point essentiel de son étude. J'ai donc cherché à voir ensuite les algorithmes existants.

2. *Algorithmes d'inférence d'interactions entre protéines*

De nombreuses personnes se sont questionnées sur le moyen d'inférer des voies ou des réseaux protéiques, en commençant par inférer les interactions entre protéines directement. Le sujet est complètement d'actualité et de nouveaux algorithmes sont implémentés régulièrement. En recherchant les mots clés « interaction inference » sur PubMed on obtient 1179 articles. Le graphe de sortie des publications nous montre bien la croissance du nombre d'articles parus chaque année : 35 en 2005, 148 en 2015.

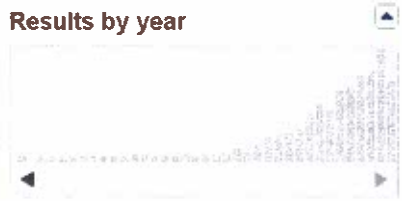


Figure 3. Evolution du nombre de parutions d'articles par an avec les mots clés « interaction inference » sur PubMed

Le premier algorithme issu d'un article que j'ai étudié est celui de *L.R.Acharya et al.* [1] et se base sur des sets d'expression de gènes. Cet algorithme considère l'inférence de pathways comme un problème d'optimisation discret. Il introduit une fonction d'énergie qui mesure l'optimalité de la structure d'une voie de signalisation.

J'ai trouvé plusieurs autres algorithmes qui utilisaient le même type de donnée en entrée : des vecteurs d'expression de gènes. Les méthodes diffèrent d'un algorithme à l'autre. Certains adoptent plutôt une approche statistique, comme ARACNE [2], MINDy [20], PathFinder [3] ou encore MI3 [21]. D'autres font plus appel à la prédiction de structure et à d'autres aspects physiques [2,7]

En faisant des recherches plus approfondies sur le type de données que ces algorithmes prenaient en entrée, je me suis rendue compte qu'il était très difficile d'obtenir des vecteurs d'expression de gènes. Il n'existe pas de base de données contenant de telles données, et les groupes capables d'en fournir en quantité sont ceux qui se concentrent sur la NGS (Next Generation Sequencing). Ce sont donc des groupes privés, avec lesquels nous n'avions pas de partenariat

3. *Algorithmes de text-mining*

En continuant mes recherches dans les publications, j'ai pu trouver un autre type d'algorithme, totalement différent des autres. Ces algorithmes permettent d'inférer des connaissances en faisant de la fouille de données textuelles. Dans le domaine scientifique, quasiment toutes traces de recherche sont disponibles dans des articles mis en ligne. Actuellement la masse de données disponible dans la littérature fait partie des plus riches et des plus intéressantes sources à exploiter. Il était donc intuitif de se tourner vers les publications scientifiques pour extraire des connaissances.

```
Algorithm: //Extraction of true PPI pairs
Input: Sentence/Abstract/Text
S: Sentence
P: Protein (P A – Protein A, P B – Protein B)
I: Relation Keyword
NE: Negation
V / N: Verbs / Nouns
IK: Root Word (corresponding to relation keyword)
HPC: Human Proteins Count
Pat: Array of patterns
Triplet pairs = PIP, PPI, IPP
Init: Exiting list = null #store extracted pairs to avoid overlap
      since one pair can satisfy more than one pattern.
Output: PubMed ID, P A, P B, I | IK, S

If Text = = Format (PPInterFinder or Medline or XML)
  Text = unique format (Text)
    For a list of S = split sentence (Text)
      S = Protein entity Regionalization (S) # NAGGNER
      S = Normalization Human protein (S) # ProNormz
      If HPC >= 2
        S = Stanford parser (S)
        S = Find V and N (S) # Using Tregex
        S = Find I (S) # IK-I dictionary
        S = Find NE (S)
        If I = = 1
          If Triplet pair = = (PIP or PPI or IPP)
            Triplet pair = possible triplets (S) # Rules set
            For each Triplet pair
              S = Tag (Triplet pair)
              While (!exact)
                Pattern Match (S, Pat) # String Pattern Matching
                If (match)
                  Declare Triplet pair
                End If
              End While
            End For
          End If
        End If
      End If
    End For
End If
```

Figure 4. Algorithme HPIminer

HPIminer [13] est l'un des premiers algorithmes traitant de text-mining dans des publications scientifiques. Cet algorithme se sert de plusieurs autres outils, notamment NAGGNER [22] pour détecter les protéines et les gènes, ProNormz [23] pour normaliser les noms des protéines et des gènes et PPInterFinder [14] pour extraire les interactions protéine-protéine. Cela m'a permis de me donner une idée un peu plus précise du déroulement d'un algorithme de text-mining. Je me suis par la suite concentrée sur PPInterFinder, qui me semblait être le point clé de leur recherche : Comment extraire des interactions à partir d'un texte ? Comment construit-on les règles de détection ? La méthode de PPInterFinder fut l'une des plus intéressantes à étudier. L'input est un article ou un abstract PubMed et l'algorithme se décompose en 3 phases : détection des mots clés de relations entre protéines, détection des pairs de protéines candidates, puis extraction de la relation en utilisant un set de 11 patterns établis. Cette méthode permet, d'après les auteurs, de prédire des interactions avec un taux d'exactitude de 66.05%.

En comparaison avec les autres types d'algorithme, celui-ci semblait donner des résultats avec une fiabilité satisfaisante, et les données en entrée sont faciles d'accès : publications scientifiques.

4. Compétiteurs

Gene Go

GeneGo Metacore, par Thomson Reuters, est une application payante, qui fournit tout type de données biologiques, notamment des pathways. Aucun partenariat n'étant en cours entre BIOVIA et Thomson Reuters nous n'avons pas pu accéder à ces données. Cependant, certains pathways étaient disponibles gratuitement en ligne. Rien n'indique officiellement comment sont générés ces pathways mais il semble que beaucoup d'informations proviennent d'articles. En effet, à chaque pathway est associé un set d'articles correspondants. Il pourrait donc s'agir de curation manuelle sur ces publications.

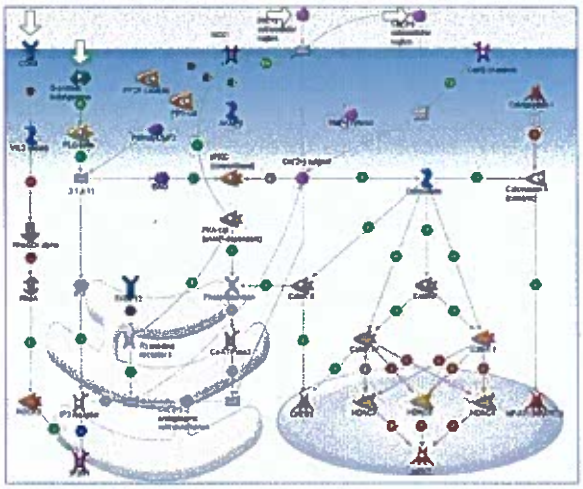


Figure 5. Exemple d'un Free Pathway GeneGo

BioNLP

BioNLP est une communauté de chercheurs se concentrant sur l'étude de PNL (programmation neurolinguistique) pour la biologie, notamment dans la littérature scientifiques. Durant leur projet de 2013 [24], ils ont démontré que l'extraction d'interaction et de relation entre protéines était possible à partir de curation dans les publications scientifiques.

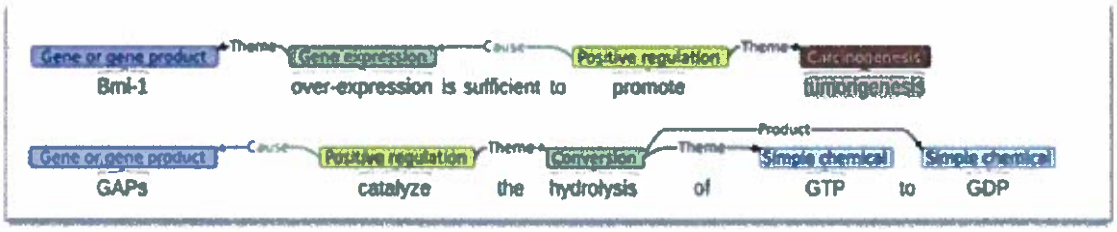


Figure 6. Schéma des annotations créées par BioNLP

Ils introduisent la notion d'annotation dans un texte et de relations entre différentes ontologies. Ces travaux de recherche ont beaucoup attiré notre attention, car on y a retrouvé des notions existantes dans la technologies qu'on prévoyait d'utiliser, et qui sera présentée dans une prochaine partie.

Pathway Studio

Pathway Studio [25] est une application créée par le groupe RELX. Elle permet de faire des études et d'extraire des données à partir de texte scientifiques, publications ou brevets. Cette application permet d'analyser et de visualiser les mécanismes d'une maladie donnée au niveau moléculaire. Elle permet

également aux utilisateurs de pouvoir comparer leurs données avec ce qui est déjà existant dans la littérature.

Après ce premier mois d'état de l'art, nous avons fait le point avec Richard pour se décider sur l'approche à privilégier. Nous avions d'une part des algorithmes nécessitant des données peu accessibles et dont les formats sont difficiles à gérer: les expressions de gènes. Et d'autre part une méthode qui nous semblait prometteuse, dont la source d'information est disponible directement et qui nous permettrait d'exploiter un outil qui sera, de toute façon, utilisé pour le produit CI : CloudView par EXALEAD. Il est également important de noter que les compétiteurs semblent tous se concentrer sur cette méthode actuellement, car les ressources dans la littérature ne font qu'accroitre depuis la dernière décennie.

C'est donc la stratégie text-mining dans les publications scientifiques que nous avons choisi d'étudier et d'approfondir. Il s'agit alors de faire une évaluation qualitative et quantitative de cette approche, et d'estimer les capacités de la technologie CloudView pour ce travail de recherche.

IV. Technologies

1. *EXALEAD-CloudView*

EXALEAD est une entreprise proposant une solution logicielle aux applications basées sur un moteur de recherche. Elle a été créée en 2000, et a lancé son premier moteur de recherche en 2006. Elle a ensuite été rachetée par DASSAULT SYSTEMES en 2010.

La solution proposée par EXALEAD est CloudView. Elle permet d'indexer une grande quantité de données afin de pouvoir les traiter et les étudier, et créer un moteur de recherche. Elle permet de customiser son pipeline d'indexation et également de personnaliser l'interface utilisateur.

La solution d'EXALEAD est notamment celle utilisée pour le site de la SNCF, et il existe un moteur de recherche internet à son nom : Exalead.com.

DASSAULT SYSTEMES utilise désormais CloudView pour quasiment tous les produits et projets nécessitant des capacités de « search » avancées et performantes. C'est donc cet outil qui a été utilisé pour notre projet, et nous verrons par la suite que nous l'avons exploité différemment de son usage initial.

Indexation by CloudView

Comprendre comment se passe une indexation avec CloudView nécessite d'assimiler toutes les notions relatives à cette technologie. Il faut également être capable de visualiser le pipeline d'indexation représenté ci-dessous :

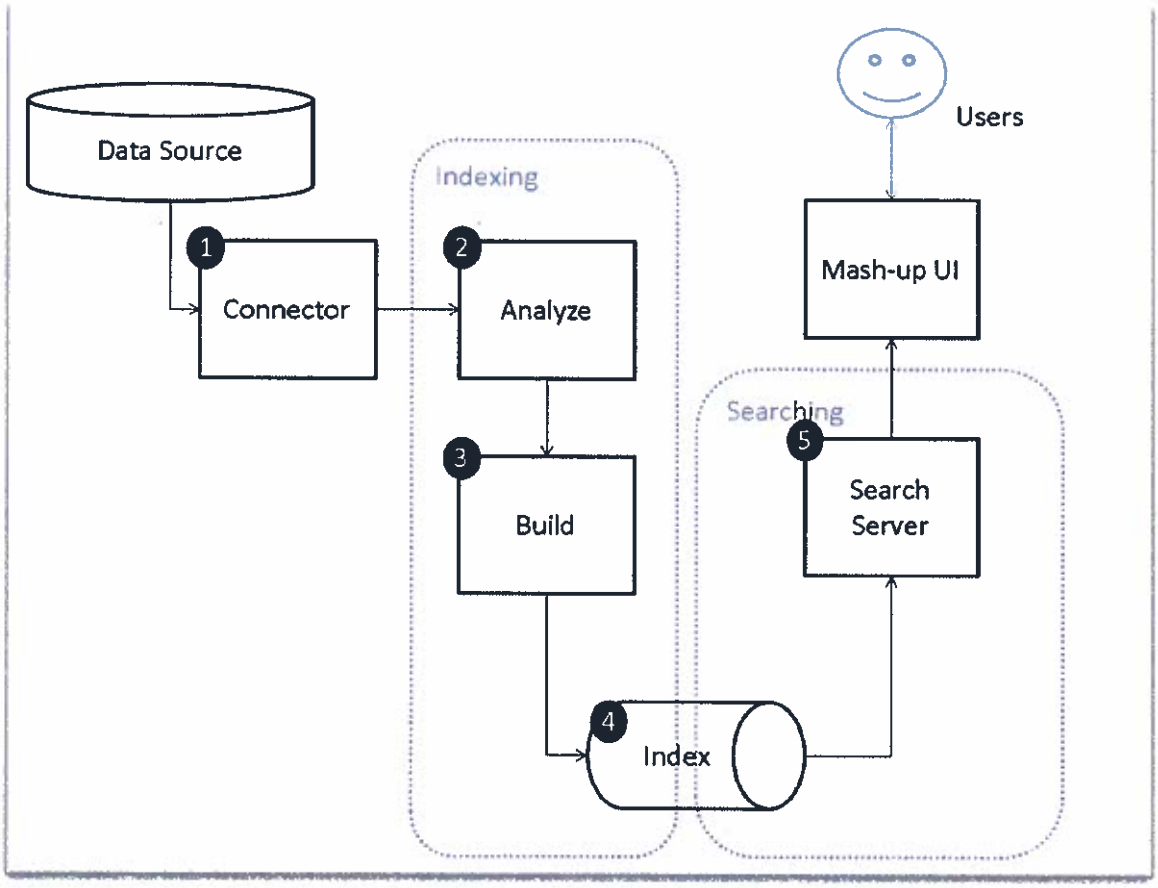


Figure 7. Schéma d'une indexation CloudView

Un des premiers concepts propre à CloudView est le Connecteur (1). C'est ce qui va servir à récupérer les données en sources pour les pousser dans le serveur en les transformant en Documents. Le connecteur est en fait un code (JAVA) contenant notamment un parser et une API permettant de pousser les données dans le serveur.

Une fois dans le serveur, les documents sont analysés (2). C'est durant cette phase d'analyse que l'on peut faire de l'extraction de connaissances et générer des *annotations*. Cette phase sera developpée dans la partie Méthodes, où j'expliciterai davantage comment fonctionnent les pipelines d'analyse et comment nous les avons utilisés.

Après l'analyse, il est possible de post-traiter les données, dans la partie Build (3). Ce concept peut servir par exemple à construire des graphes à partir de données indexées. Tout sera également explicité plus en détail dans la partie Méthodes.

Une fois le traitement de données terminé, les documents sont placés dans l'Index (4), puis répliqués dans le serveur de Search (5). C'est par ce concept que les utilisateurs pourront rechercher leur informations dans l'index, à l'aide de requêtes UQL (Universal Query Language) [18].

2. JAVA

Nous avons également utilisé le langage de programmation JAVA afin d'implémenter des méthodes pour post-traiter nos données. En effet, CloudView ne nous a pas servi à faire l'entière étude, et nous avons donc eu besoin de récupérer, vérifier, comparer et visualiser nos données à l'aide d'un autre programme. Tout a été fait sur Eclipse.

3. CytoScape

Cytoscape [26] est logiciel libre d'accès qui permet de visualiser des réseaux biologiques. Il nous a servi à visualiser les interactions protéine-protéine qu'on a pu extraire des publications scientifiques.

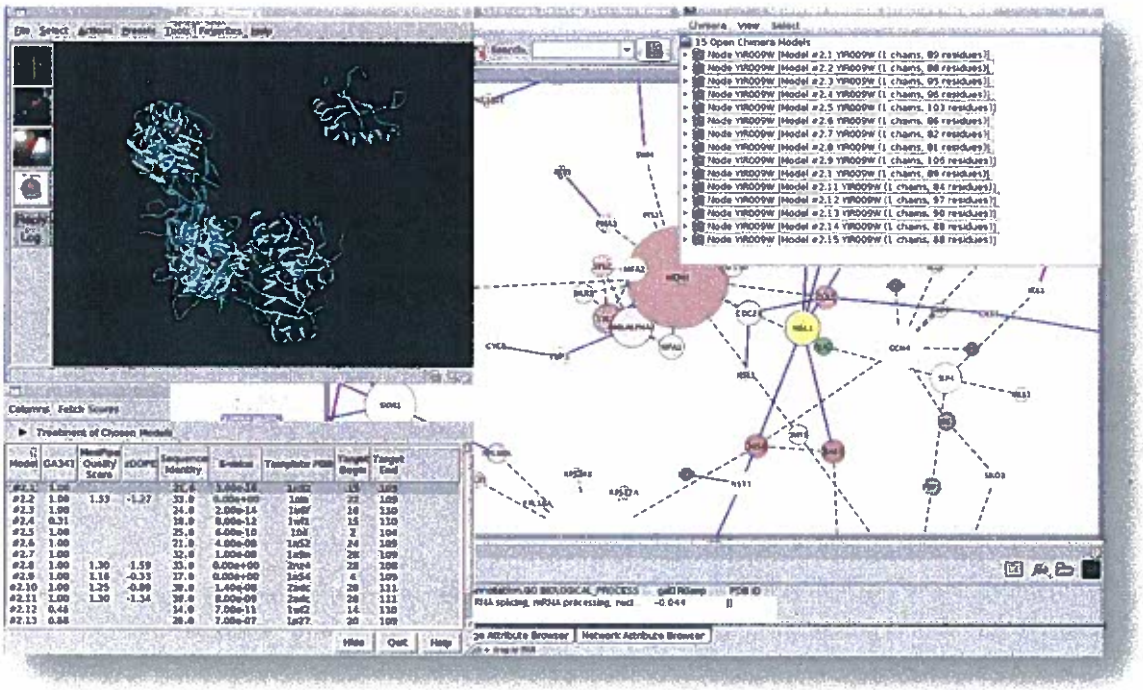


Figure 8. Exemple d'une utilisation Cytoscape

V. Data Sources

Avant d'entrer dans les détails des méthodes que nous avons implémentées, il est nécessaire d'introduire les sources dont nous avons eu besoin. Voici donc les 5 principales sources de données que nous avons utilisées, toutes libres d'accès et gratuites :

Pubmed

Nous avons choisi de prendre en entrée des publications scientifiques pour implémenter notre méthode de text-mining. Une des plus grandes bases de données en termes de publications est celle fournie par le NCBI : PubMed. Elle compte aujourd'hui plus de 4 millions de publications scientifiques, avec une croissance continue d'entrées par année. Elle est gratuite et très régulièrement mise à jour, c'est donc avec ce corpus d'articles que nous avons décidé de travailler.

Cependant il est important de noter que tous les articles ne sont pas forcément entièrement libres d'accès. Pour pallier à ce problème, nous avons donc décidé de n'étudier que les abstracts (extraits), qui eux sont disponibles pour toutes les publications.

IntAct

IntAct [11] est une base de données publique et ouverte aux utilisateurs, de sorte que chaque chercheur peut y entrer l'interaction qu'il a trouvé. Pour chaque entrée, des informations complémentaires sont données, entre autre le type d'expérience ayant permis de découvrir l'information, et la ou les publication(s) associée(s). IntAct contient environ 900 000 interactions en 2016.

Uniprot

Uniprot [8] est également une des plus grandes bases de données dans son domaine : les protéines. Elle compte près de 67 millions de protéines, départagées en deux sous bases de données : TrEMBL et SwissProt, SwissProt étant la partie contenant les données vérifiées manuellement. Cette distinction n'a pas été un critère de sélection pour notre étude. Cependant, nous n'avons pris que les protéines humaines pour créer notre dictionnaire.

NCBI Gene

NCBI Gene [9] est la base de données qui nous a servi à créer notre ontologie de gènes. Tout comme pour les protéines, les gènes ont une identifiant unique dans cette base de données, qui nous a servi à normaliser les formes retrouvées dans les publications.

Disease Ontology

La Disease Ontology [10] est un dictionnaire de maladies, regroupées par synonymes, avec leur forme normalisée. Elle nous a donc servi à détecter et annoter les maladies dans les articles.

VI. Méthodes

La conclusion de mon état de l'art fut que la méthode qui semblait la plus prometteuse était l'approche Text-Mining. Nous avons choisi, pour l'implémenter, d'utiliser la technologie CloudView. Nous allons voir à présent comment CloudView a été exploité pour extraire des interactions entre protéines dans des publications.

1. *Indexation*

La source de données en entrée est le corpus d'abstracts provenant de PubMed. Ce sont donc une partie de ces données qui ont été indexées à l'aide de CloudView. Comme cela a été expliqué dans la partie Technologies, un *connecteur* se charge d'extraire et de transformer les données sources en *documents*. Le connecteur permet également de créer des *« Méta-datas »*. Ces métas-datas sont des objets qui définissent le document, en schéma UML l'équivalent serait les attributs, ou propriétés.

Ici, les métas-data générées par le connecteur sont par exemple le titre, les auteurs, l'abstract, mais également les protéines détectées lors de l'analyse de l'abstract ou les patterns. Nous verrons comment la phase d'analyse permet de générer des métas-datas dans la partie suivante.

2. *Détection d'entités nommées*

> *Ontologie :*
>
> *Ensemble structuré des termes et concepts représentant le sens d'un champ d'informations, que ce soit par les métadonnées d'un espace de noms, ou les éléments d'un domaine de connaissances. L'ontologie constitue en soi un modèle de données représentatif d'un ensemble de concepts dans un domaine, ainsi que des relations entre ces concepts. Elle est employée pour raisonner à propos des objets du domaine concerné. Plus simplement, on peut aussi dire que l' « ontologie est aux données ce que la grammaire est au langage ».* (Source : Wikipédia)

Une fois les documents générés par le connecteur, ils sont analysés. Cette analyse se fait à l'aide d'un pipeline composé de processeurs ou d'autres pipelines imbriqués. Chaque processeur permet une analyse différente du texte. Un processeur peut par exemple détecter le langage utilisé dans un texte, ou encore détecter la nature des mots (verbes, noms, adjectifs etc...). Certains permettent également d'annoter du texte ; dépendant de la façon dont on a configuré le processeur une certaine annotation est générée. Ces annotations peuvent ensuite être stockées dans des métas-datas si besoin.

Pour mon travail de recherche j'ai repris une instance de CloudView qui avait été préconfigurée. Les éléments déjà présents étaient les processeurs par défaut, entre autres le détecteur de langage ou le détecteur de date.

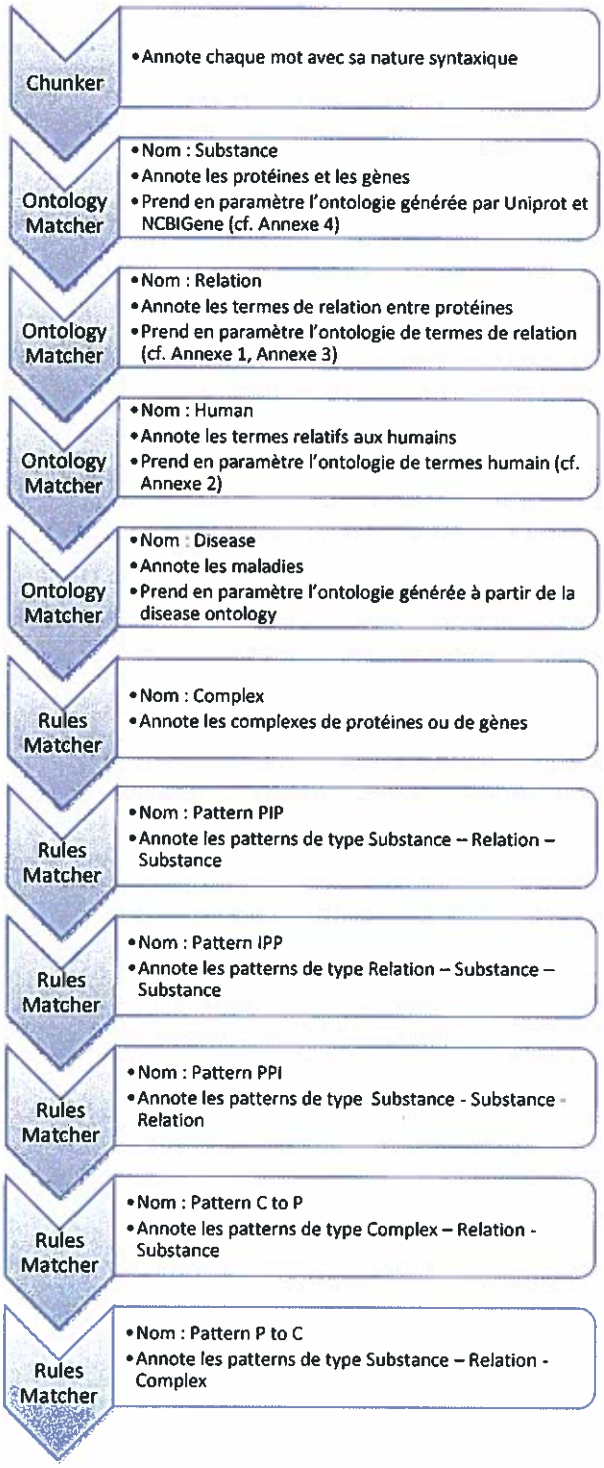


Figure 9. Schéma du pipeline sémantique

J'ai pu pour ma part introduire un pipeline appelé « Semantic Pipeline » (cf. Figure 9). Ce pipeline intégré dans le pipeline d'analyse est celui qui contient tous les processeurs permettant de détecter et d'annoter les objets qui nous intéressent.

Pour ce dont nous avions besoin, 4 types de processeurs ont été utilisés :

1. Chunker: Permet d'annoter chaque mot avec leur nature syntaxique (verbe-nom-adverbe-adjectif)
2. Ontology Matcher : Permet de détecter tous les mots présents dans un dictionnaire, une ontologie passée en paramètre, et de les annoter.
3. Rules Matcher : Permet de créer des règles afin de détecter et d'annoter des patterns.
4. Java Document Processor : Permet d'introduire un petit code JAVA, on expliquera plus tard en quoi ce processeur nous a servi.

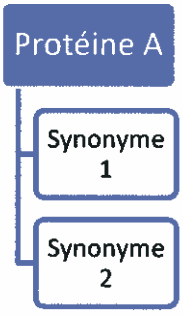
L'ordre dans lequel les processeurs sont placés dans le pipeline a son importance car une annotation générée par un Ontology matcher peut par exemple servir à la détection d'un pattern avec un Rules Matcher.

Les ontologies « Substance » et « Disease » proviennent de sources publiques (voir Data Sources). Mais j'ai beaucoup travaillé sur l'ontologie de termes de relation. J'en avais une première version à la main (cf Annexe 1). Elle était très faible et ne me convenait pas. En me replongeant dans la littérature, j'ai pu en trouver une beaucoup plus riche, provenant d'une étude [14], et fournie en donnée supplémentaire. Elle ne contient pas seulement des termes de **régulation** mais de **relation** entre protéines et gènes.

A la sortie de chaque Ontology Matcher, des annotations sont générées. Les annotations comme les substances ou les termes de relation vont servir à configurer les règles permettant de détecter les patterns.

Une ontologie est toujours organisée de façon à ce que chaque élément ait une forme normalisée. Les annotations générées sont alors toujours sous la forme normalisée.

Pour illustrer, si mon ontologie de Substance contient :



Où Protéine A est la forme normalisée.

Si la phrase de l'article est :

Synonyme 2 active Protéine B

Alors l'annotation générée pour *Synonyme 2* sera Protéine A.

Figure 10. Exemple d'un élément de l'ontologie Substance

A la fin du pipeline d'indexation, et donc en dehors du pipeline sémantique, j'ai ajouté un JAVA Document Processor (voir annexe). Ce processeur permettant de créer de petit code JAVA, m'a servi à créer des booléens qui permettent de savoir si un abstract contient un terme humain ou non, contient des substances ou non, contient des maladies ou non, contient des patterns ou non. Ainsi des métas-data « is human », « has substance », « has disease », ou encore « has pattern » sont générés pour chaque abstracts et ont servi à la récupération des données dans le code JAVA fait sur Eclipse.

3. *Création de règles*

Nous allons à présent étudier les règles plus en profondeur. C'est en effet sur cette partie que j'ai le plus travaillé. Cette recherche de patterns est le centre de mon étude. C'est grâce à ces règles que l'extraction de connaissances devient possible. Cependant, il fallait que mes règles soient capables de détecter un maximum de vraies informations, sans détecter trop de faux positifs : maximiser la sensibilité tout en minimisant la spécificité, là était toute la problématique.

Cette phase de réflexion a pris une grande partie de mon temps et m'a amenée à échanger avec les responsables de la partie « Semantic Factory » chez EXALEAD. Plusieurs articles m'ont également beaucoup aidée dans ma réflexion, plus particulièrement celui présentant le projet BioNLP 2013, présenté dans l'état de l'art [24].

Les règles – Rules Matcher – se créent et se configurent à l'aide d'une interface graphique (cf. Figure 11).

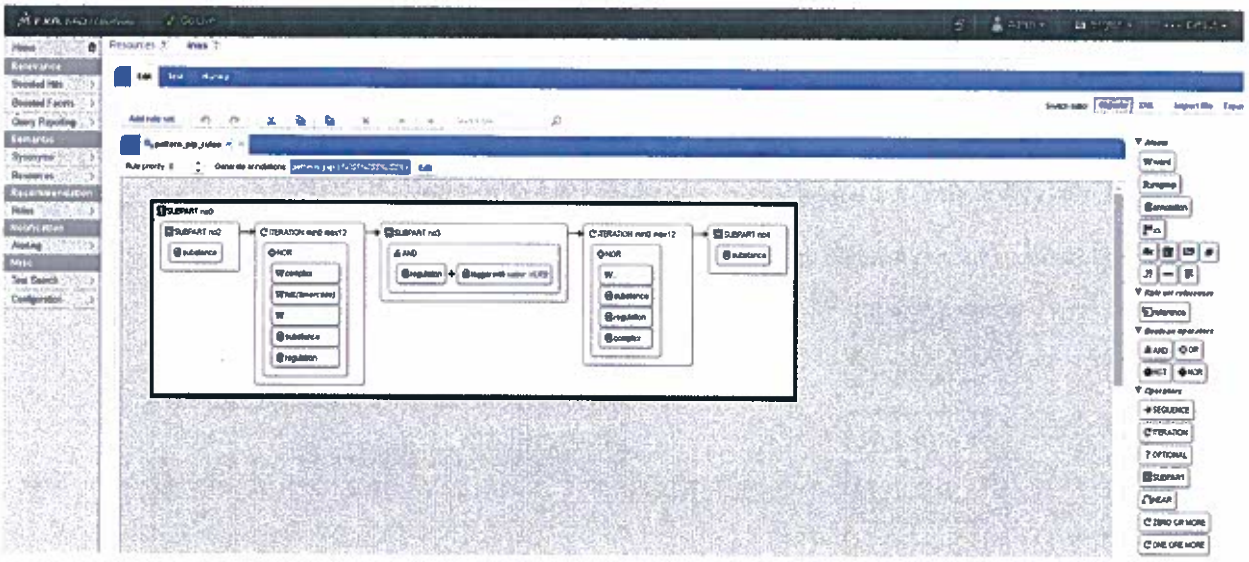


Figure 11. Interface graphique de création de règles (Rules Matcher)

Comme on peut le voir sur le pipeline sémantique, six Rules Matcher ont été configurés.

Nous allons expliciter ces règles, et pour chacune d'entre elles, donner un exemple de ce qu'on peut récupérer comme informations.

Note : Les phrases données en exemple sont biologiquement fausses. Le but est d'illustrer le principe de chaque règle.

Règle 1 : Complex

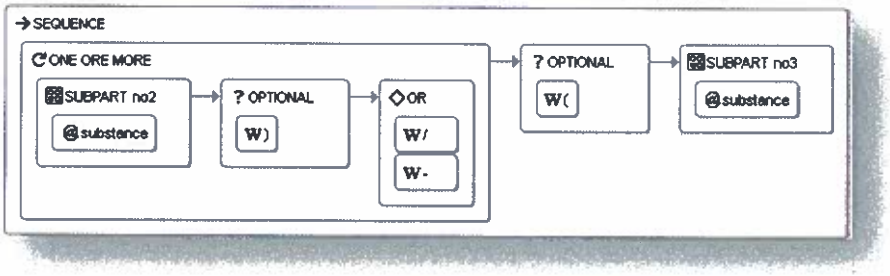


Figure 12. Règle de détection des complexes

Ici, la règle est une séquence d'une ou plusieurs Substance (Annotation générée par l'Ontology Matcher Substance) suivie d'un / ou d'un – et finissant par une Substance. Les parenthèses sont optionnelles car il arrive qu'un complexe soit de la forme (Substance)-Substance, ou encore Substance/(Substance), notamment lorsque le nom de la substance est composé.

Cette règle génère elle-même une annotation, appelée Complex, qui servira à configurer les 2 dernières règles. Cette annotation est donc l'ensemble des substances formant le complexe détecté, sous leur forme normalisée.

Les « SUBPART » numérotés dans chaque règle servent à gérer les annotations générées. Ici (cf. Figure 12), on attribue un numéro à chaque annotation Substance afin que l'annotation ne contienne que le nom des substances (normalisées).

Exemple:

Règle 2: Pattern PIP

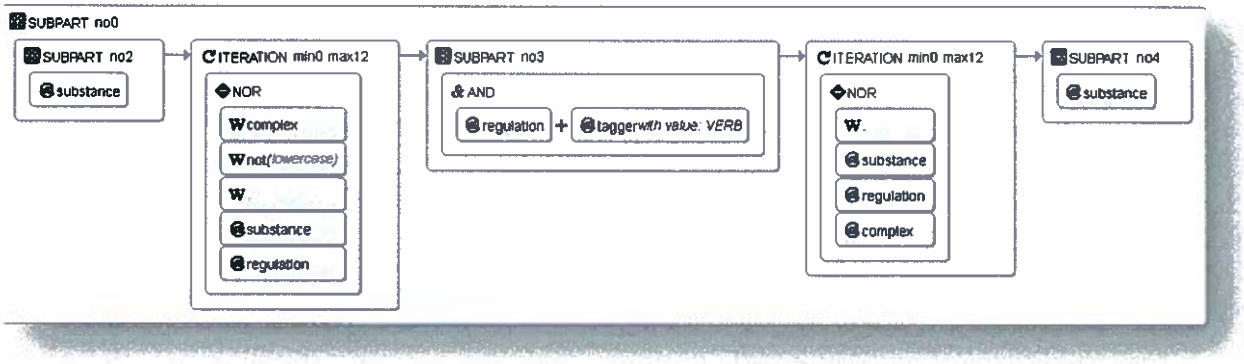


Figure 13. Règle de détection des patterns de type Substance-Relation-Substance

Cette règle est une des plus importante, voire la plus importante, car la majorité des interactions sont décrites suivant ce pattern dans les textes. Cependant, il a fallu revoir cette règle à plusieurs reprises pour en arriver à ce stade.

Tout d'abord, il fallait se soucier du problème de négation. D'où l'exclusion du mot « not » entre la première substance et le terme de relation, bien que cela reste insuffisant. De plus, mon ontologie de terme de relation (cf. Annexe 3) contient aussi bien des noms que des verbes. Pour ce type de pattern, il fallait absolument que je limite le terme de relation à un verbe. C'est donc ici que me sert le processeur « Chunker » vu précédemment, qui a annoté tous les mots avec leur nature syntaxique. Le terme de relation (appelé «regulation» dans la règle) est donc forcé d'être un verbe. Il fallait aussi faire attention au fait que le pattern ne chevauche pas deux phrases différentes, ce qui n'aurait aucun sens. J'ai donc interdit les «.» entre les différents objets.

Pour résumer, et pour mieux comprendre les autres règles, ici la règle implémentée (cf. Figure 13) veut dire qu'on recherche un pattern du type :

Un mot annoté comme une substance + une itération de 0 à 12 mots ne contenant pas de complexe, de substance, de terme de relation, de point et de « not » + un terme de relation étant annoté comme un verbe + une itération de 0 à 12 mots restreints + un mot annoté comme une substance.

L'annotation généré par cette règle contient : Le pattern (Subpart 0) + la première substance (Subpart 2) + le terme de relation (Subpart 3) + la deuxième substance (Subpart 4).

Pour toutes les autres règles les annotations générées seront de la même forme :

PATTERN – INTERACTEUR 1 – RELATION – INTERACTEUR 2

Exemple :

In that context, EGFR is activated by EGF and not by APB.

Règle 3: Pattern IPP

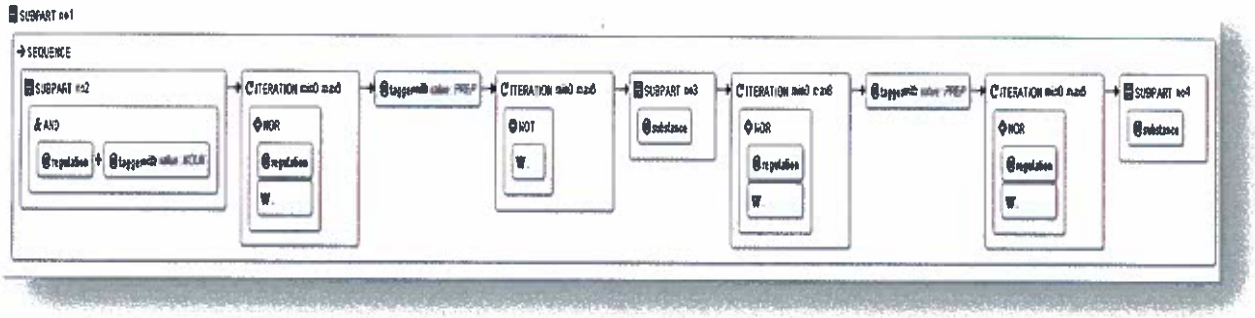


Figure 14. Règle de détection de patterns de type Relation-Substance-Substance

D'après nos tests et les résultats trouvés, ce type de pattern est moins présent que le pattern PIP. Néanmoins, il reste important et m'a également demandée un grand travail de réflexion. Tout comme la règle précédente, il fallait se soucier de la nature du mot Relation. Ici, on recherche des phrases ou des expressions commençant par le terme de relation. On ne veut donc pas de verbe mais plutôt un nom. C'est donc ce qui a été fixé. Les restrictions pour les itérations de mots entre les annotations sont restées les mêmes. Cela permet d'une certaine façon d'éviter que les règles se chevauchent.

Exemple:

In another context, it was proved that the activation of EGFR by EGF is true.

Règle 4: Pattern PPI

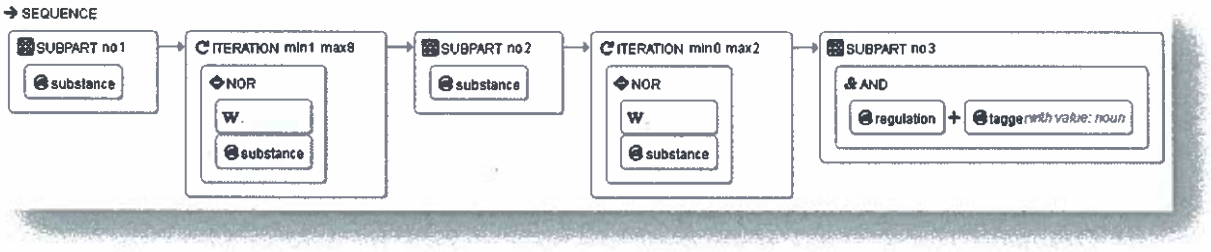


Figure 15. Règles de détection de patterns de type Substance-Substance-Relation

Cette règle était une suite logique aux deux premières règles implémentées. Cependant, j'ai beaucoup hésité à la garder car elle génère plus de faux positifs que de vraies informations. En effet, il est rare que les phrases décrivant une interaction soient de la forme Substance – Substance – Relation.

Mon but était de récupérer des patterns comme celui présent dans l'exemple ci-dessous :

Exemple:

This research aimed to prove that EGF and EGFR are binding together.

Après une étude des résultats de tests, j'ai finalement choisi de ne pas la garder.

Règle 5: Pattern Complex to Substance

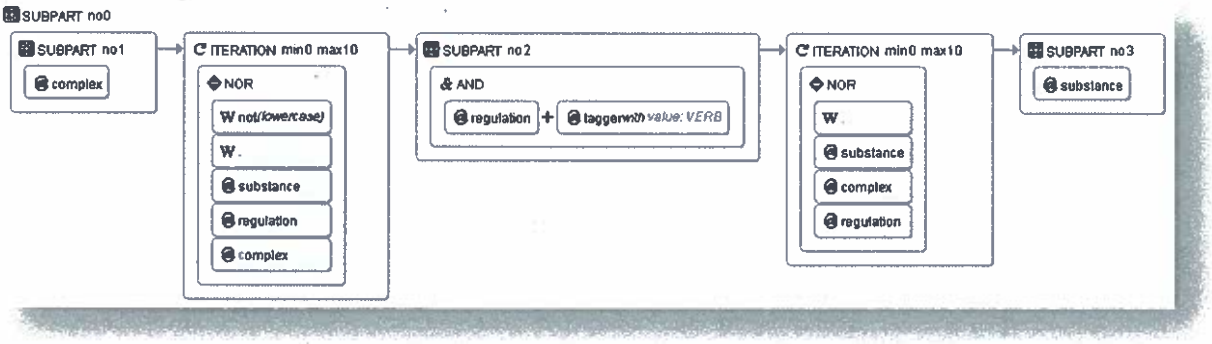


Figure 16. Règle de détection de patterns de type Complexe-Relation-Substance

J'ai configuré deux règles impliquant les complexes car les substances sont souvent impliquées dans des complexes. Cela générait beaucoup de faux positifs pour la règle de pattern PIP. Ces règles m'ont permises de traiter ces patterns indépendamment des interactions impliquant des substances seules.

Exemple :

In the human cells, the EGF/EGFR complex is inhibited by PKC.

Règle 6: Pattern Substance to Complex

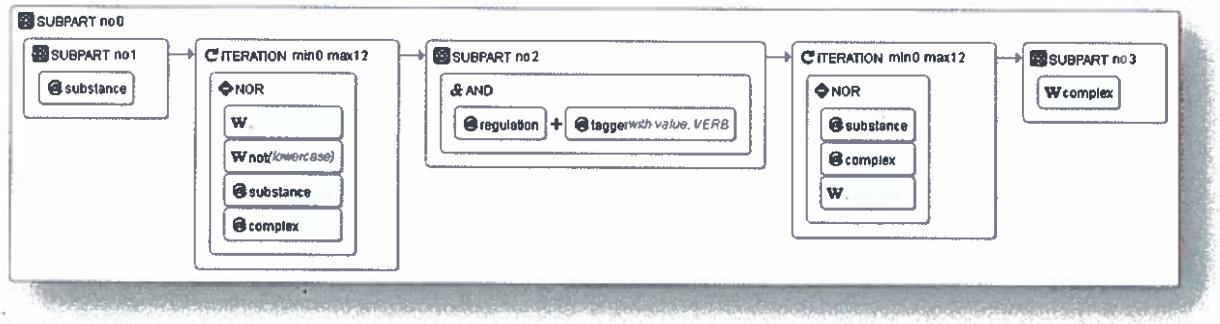


Figure 17. Règle de détection de pattern de type Substance Relation-Complexe

Exemple :

We proved that PKC negatively regulates the EGF-EGFR complex.

4. *Récupération des données*

Une fois l'indexation terminée toutes les métas-datas sont disponibles et accessibles. Afin d'analyser les résultats, nous avons choisi d'utiliser un programme codé en JAVA (voir annexe). Ce programme permet de récupérer les métas-datas intéressantes et de les post-traiter. Ce programme permet également de générer un fichier CSV du type :

PATTERN	INTERACTOR 1	RELATION	INTERACTOR 2

Table 1. Modèle du fichier CSV généré

La base de ce programme JAVA a été codée par Frédéric. Cette base contenait initialement la méthode pour se connecter au serveur CloudView et une méthode pour compter le nombre de patterns détectés. J'ai ensuite travaillé sur ce code pour récupérer les patterns, de tout type, ainsi que les « interactors » (protéines et gènes) et les termes de relations associés. J'ai également codé la méthode permettant de générer le fichier CSV.

5. Vérification et Réflexion

L'étape de vérification a chevauché l'étape de création de règles car elle m'a servie justement à ajuster et à améliorer ces règles. J'ai commencé par me constituer un petit set d'articles pour réaliser mes tests. Ce set d'articles n'a pas été choisi au hasard : il provient d'un des free pathways disponibles sur GeneGo. Comme nous l'avons mentionné dans l'état de l'art, ces pathways disponibles gratuitement sont toujours associés à une série d'articles décrivant les interactions qui le composent. J'ai donc choisi un des pathways (cf. Annexe 5) et indexé tous les articles qui lui sont associés (cf. Annexe 6). Ce jeu de test m'a permis de comparer les interactions extraites des articles de celles présentes dans le pathway. C'est à ce moment que j'ai pu constater que les patterns extraits des articles ne décrivaient pas forcément (et même très rarement) des interactions directes.

D'un autre côté, nous avions déjà choisi de prendre la base de données IntAct comme référence pour nos vérifications. Les interactions présentes dans cette base de données sont pour la plupart des interactions directes et binaires. Nous avons donc décidé de récupérer cette base de données et de la réindexer sur une autre instance de CloudView. Le but de cette indexation était d'avoir à portée de main une base de données d'interactions binaires, et de pouvoir construire des chemins (paths) entre deux substances à partir de celles-ci. Ceci est tout à fait possible avec CloudView ; dans la partie « build » (cf. Figure ***) il est possible de post-traiter les données, notamment avec la partie « consolidation » de la console d'administration (cf. Annexe 8). Cette fonctionnalité de CloudView permet de générer des paths à partir de données binaires, en paramétrant par exemple le nombre maximum de couple qui le compose (longueur du path). Ces informations nous permettent une vérification supplémentaire sur les interactions extraites des abstracts. En plus de savoir si l'interaction est présente ou non dans IntAct, on peut maintenant déterminer s'il existe un path entre les deux acteurs de l'interaction (dans le cas où l'interaction n'est pas directe). Nous avons donc attribué deux scores à chaque interaction : l'existence dans Intact et l'existence d'un path de longueur 5 entre les deux substances composés d'interactions binaires IntAct.

PATTERN	INTERACTOR 1	RELATION	INTERACTOR 2	SCORE INTACT	SCORE PATH

Table 2. Modèle du fichier CSV généré avec les scores

6. Génération de graphes

A partir d'un fichier CSV, il est possible de construire un graphe sur l'application Cytoscape. Dans la partie Technologies, nous avons brièvement vu en quoi consistait cet outil. Ici nous l'avons utilisé pour visualiser les interactions détectées avec un score IntAct non nul.

VII. Résultats et Discussion

L'approche Text-mining sur des publications scientifiques pour l'inférence d'interactions entre protéines et de réseaux biologiques était celle qui semblait la plus « sure ». A l'aide de la solution CloudView par EXALEAD, nous avons pu extraire des données relatives aux interactions entre protéines dans les abstracts PubMed. En utilisant cette même technologie nous avons pu réutiliser une base de données connues en termes d'interactions entre protéines : IntAct. Nous avons ainsi généré des chemins composés d'interactions binaires entre deux substances. C'est ce qui nous a permis de donner un score à nos données extraites de la littérature.

Nous avons donc montré que l'outil CloudView est un réel atout pour une étude sur l'inférence d'interactions de protéines et de réseaux biologiques.

Dans cette partie, je vais donc expliciter les résultats à l'aide d'un exemple concret, et montrer comment passer d'un set d'articles scientifiques en entrée à un graphe d'interactions entres protéines en sortie. Nous étudierons ensuite ces résultats pour parler des limites et des perspectives de mon travail de recherche.

1. *Exemple et résultats*

Pour étudier nos résultats, je me suis concentré sur une protéine bien connue en biologie : EGFR (Epidermal Growth Factor Receptor). Pour cela, nous avons indexé une partie des abstracts de publications PubMed. Une fois l'indexation terminée nous avons récupéré la liste des interactions détectés avec notre fichier CSV (cf. Annexe 11).

Le fichier contient 793 patterns contenant la protéine EGFR et voici les résultats:

Nombre de patterns ayant un Score Intact Non Nul	Nombre de pattern ayant un Score Path Non Nul
50 6.3%	585 73.7%

Table 3. Tableau des scores avec l'exemple EGFR

Nous avons également effectué cette étude de comparaison dans l'autre sens. Nous avons récupéré toutes les interactions binaires impliquant la protéine EGFR dans IntAct (qui sont donc des interactions directes) et nous avons regardé si ces interactions été détectées dans nos abstracts PubMed. Cette étude a été faite grâce à une méthode implémentée dans notre code JAVA. Elle permet de générer un autre fichier CSV (cf. Annexe 12) contenant toutes les interactions d'IntAct avec EGFR et, pour chacune d'entre elles, indiquer le nombre de fois où elle apparait dans IntAct et le nombre de patterns détectés la contenant.

Le fichier contient 485 interactions IntAct impliquant EGFR, voici les résultats :

Nombre d'interaction n'étant présentes qu'une seule fois dans IntAct	Nombre d'interactions retrouvées dans les abstracts PubMed Indexés
319 65%	50 10%

Table 4. Tableau de comparaison avec les interactions d'IntAct

On voit donc que peu d'interactions sont communes entre la base de données IntAct et ce qu'on extrait des abstracts. Une des explications possibles est, comme nous l'avons remarqué précédemment, que les patterns extraits des textes décrivent rarement des interactions directes. Cette hypothèse est très fortement confirmée par le fait que plus de 70 % des patterns mettent en relations des substances pour lesquelles il existe un chemin d'interactions directes (cf. Tableau 3).

Une fois les fichiers CSV générés, nous avons voulu visualisé les données.

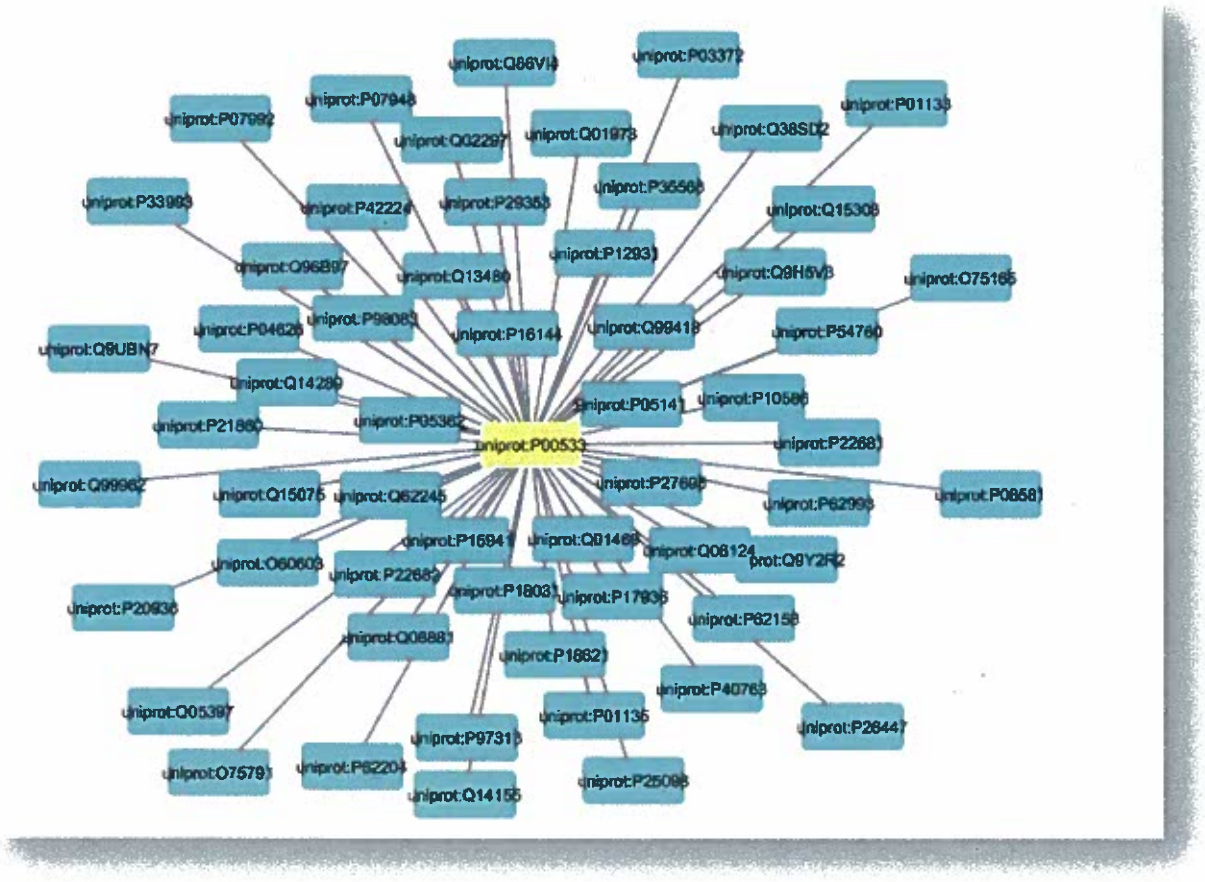


Figure 18. Graphe d'interactions avec EGFR ayant un score IntAct non nul, sur Cytoscape

Le graphe présenté dans la figure 18 montre toutes les interactions binaires avec EGFR extraites des abstracts PubMed indexés et existants dans la base de données IntAct.

Nous aurions voulu affiché un graphe, avec des interactions de couleurs différentes, dépendant de leur score IntAct ou Path. Le temps m'a manquée pour réaliser ce travail. Il aurait également été intéressant de visualiser les paths générés grâce à l'indexation d'IntAct.

2. Limites

Comme nous l'avons vu, les résultats obtenus sont satisfaisants, mais ne garantissent pas une fiabilité absolue, ni même l'optimisation du rapport spécificité/sensibilité. La base de mon travail de recherche, et ce qui est au centre de toute cette étude, est la création de règles permettant de détecter les interactions dans des données textuelles. Pour se faire, j'ai pu demander l'avis d'autres collaborateurs, et me baser sur des publications, mais je n'ai pas vraiment pu échanger avec un expert en PNL (Programmation Neurolinguistique) ce qui m'aurait surement beaucoup aidée. Je suis en effet consciente que la création de ces règles nécessite d'une part, une bonne connaissance de la biologie et des relations entre protéines ou gènes, et d'autre part, une connaissance en analyse syntaxique.

Durant chaque phase de test, de nouvelles données m'amenaient à effectuer d'autres modifications dans mes règles. Le temps m'étant limité, j'ai dû me fixer une date limite à partir de laquelle je fixais les règles, afin de poursuivre sur d'autres taches. J'ai donc dû m'arrêter dans leur amélioration malgré le fait que je sois consciente des problèmes toujours existants. Je pense par exemple aux problèmes de superposition de règles sur une même phrase. En effet, je n'ai pas pu créer de réel système d'exclusivité pour chaque règle. Mais j'ai pu en discuter avec un collaborateur d'EXALEAD R&D qui m'a spécifiée qu'il existait un moyen d'attribuer une certaine priorité à chaque règle afin qu'elles ne se superposent pas.

Il est également important de noter que l'intégralité de ce travail de text-mining a été effectuée sur les abstracts seulement. PubMed fournit gratuitement tous les abstracts, mais il devient plus difficile de récupérer les articles entiers. L'approche n'aurait d'ailleurs pas été la même si l'étude avait portée sur des publications détaillées : les phrases étant plus simplifiées dans l'abstract.

En termes de visualisation des données, je n'ai pas poussé les recherches d'outils existants très loin. En voyant que Cytoscape était libre d'accès, et assez simple d'utilisation, je me suis tout de suite positionnée sur lui. Un autre outil pourrait être meilleur, esthétiquement parlant, mais aussi en ce qui concerne l'analyse de graphe.

D'autres limitations sont à mentionnées, au niveau des ontologies. Les ontologies Substances et Disease, telles qu'elles sont générées actuellement, extraient de nombreux faux positifs. Le problème des acronymes n'a pas été résolu et les mots qui ne sont pas des diseases ou des substances sont souvent annotés comme telles.

L'ontologie de termes de relation est, elle, très riche mais pose des problèmes d'ambiguïté. Une autre phase de « raffinement » permettrait d'extraire des patterns de plus en plus précis.

3. *Perspectives*

Une fois l'approche text-mining validée après mon premier mois de stage, j'ai beaucoup réfléchi aux options qui se présentaient à moi pour ce type de recherche. Cette approche ouvre des horizons de réflexion qui méritent tous à être étudiés. Je pense particulièrement à tout ce qui attrait à la Data Science/Machine Learning. J'ai notamment envisagé le fait de générer mes règles à partir d'un jeu d'apprentissage composé d'abstracts. Cela m'aurait permis de mieux identifier les tournures de phrases utilisées par les auteurs. Il aurait aussi été intéressant de pouvoir comparer les règles générées à partir d'un apprentissage statistique de celles qu'on a configurées, et mieux encore, de comparer les résultats obtenus avec ces deux méthodes.

En ce qui concerne la génération de graphe, nous avons malheureusement manqué de temps pour générer un graphe d'interactions directes uniquement. J'aurai voulu étudier plus en profondeur les paths qu'on a générés à partir de l'indexation d'IntAct, en utilisant des propriétés d'analyse de graphes, comme les algorithmes de plus court chemin par exemple.

En échangeant avec les collaborateurs d'EXALEAD, ils nous ont fait part d'un projet en court de conception, nommé APOLLO, qui consisterait à intégrer les fonctionnalités d'indexation et d'analyse de graphes dans CloudView. Une suite logique à mon étude serait d'utiliser cette nouvelle version pour analyser les graphes générés.

Conclusion

Pour conclure, mon travail a servi de preuve de faisabilité d'une part dans le type d'approche adoptée pour l'inférence d'interactions entre protéines et de réseaux biologiques, et d'autre part de l'outil utilisé pour réaliser ce projet.

Le text-mining est une méthode prometteuse, qui permet d'obtenir des résultats qualifiables et dont les données en entrée sont faciles à manipuler, et surtout très riches. Cette étude de fouille de données dans les abstracts PubMed n'est pas la première et sera surement loin d'être la dernière, mais les règles implémentées font que chaque étude sera unique. Ce travail de recherche n'a fait qu'accroitre ma curiosité dans le domaine, et c'est dans ce secteur que j'ai basé mes recherches pour mes projets futurs.

Travailler au sein d'une grande société m'a également permis de comprendre que les contraintes ne nous permettent pas toujours d'avancer au rythme voulu. Les technologies sont souvent imposées, ce qui peut être un atout, mais cela nous oblige à réfléchir notre projet à l'image du groupe, et donc parfois à nous limiter.

Ce sujet de recherche m'a beaucoup apportée et je ferai en sorte que mes compétences en CloudView EXALEAD me servent dans mes futurs projets.

Bibliographie

1. Lipi R. Acharya, Thair Judeh, Guangdi Wand and Dongxiao Zhu, Optimal structural inference of signaling pathways from unordered and overlapping gene sets, 2011
2. Adam A Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, Andrea Califano, ARACNE : An algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context, 2004
3. Gurkan Bebek, Jiong Yang, PathFinder: mining signal transduction pathway segments from protein-protein interaction networks, 2007
4. Jakob Dohrmann, Juris Puchin, Rahul Singh, Global multiple protein-protein interaction network alignment by combining pairwise network alignments, 2015
5. Surabhi Maheshwari, Michal Brylinski, Prediction of protein-protein interaction sites from weakly homologous template structures using meta-threading and machine learning, 2014
6. Rohit Singh, Daniel Park, Jinbo Xu, Raghavendra Hosur, Bonnie Berger, Struct2Net : a web service to predict protein-protein interactions using a structure-based approach, 2010
7. Anthony Gitter, Judith Klein-Seetharaman, Anupam Gupta, Ziv Bar-Joseph, Discovering pathways by orienting edges in protein interaction networks, 2010
8. The UniProt Consortium, Uniprot : a hub for protein information, 2014
9. Donna Maglott, Jim Ostell, Kim D. Pruitt, Tatiana Tatusova, Entrez Gene : gene-centered information at NCBI, 2011
10. Lynn Marie Schriml, Cesar Arze, Suvarna Nadendla, Yu-Wei Wayne Chang, Mark Mazaitis, Victor Felix, Gang Feng, Warren Alden Kibbe, Disease Ontology : a backbone for disease semantic integration, 2011
11. Hermjakob H, Montecchi-Palazzi L, Lewington C, Mudali S, Kerrien S, Orchard S, Vingron M, Roechert B, Roepstorff P, Valencia A, Margalit H, Armstrong J, Bairoch A, Cesareni G, Sherman D, Apweiler R, IntAct : an open source molecular interaction database, 2004
12. Christian von Mering, Lars J. Jensen, Michael Kuhn, Samuel Chaffron, Tobias Doerks, Beate Kruger, Berend Snel, Peer Bork, STRING 7 – recent developments in the integration and prediction of protein interactions, 2007
13. Suresh Subramani, Raja Kalpana, Pankaj Moses Monickaraj, Jeyakumar Natarajan, HPIminer: A text mining system for building and visualizing human protein interaction networks and pathways, 2015
14. Kalpana Raja, Suresh Subramani, Jeyakumar Natarajan, PPInterFinder – a mining tool for extracting causal relations on human proteins from literature, 2013
15. Fei Zhu, Preecha Patumcharoenpo, Cheng Zhang, Yang Yang, Jonathan Chan, Assawin Meechai, Wanwipa Vongsangnak, Bairong Shen, Biomedical text mining and its applications in cancer research, 2012
16. Hagit Shatkay, Scott Brady, Andrew Wong, Text as data: Using text-based features for proteins representation and for computational prediction of their characteristics, 2015

17. Sune Pletscher-Frankild, Albert Palleja, Kalliopi Tsafou, Janos X. Binder, Lars Juhl Jensen, DISEASES : Text mining and data integration of disease-gene associations, 2015
18. M. N. Grinev, S. D. Kuznestov, UQL : A UML-based Query Language for Integrated Data, 2002
19. Sveltlana Novichkova, Sergei Egorov, Nikolai Daraselia, MedScan, a natural language processing engine for MEDLINE abstracts, 2003
20. Kai Wang, Masumichi Saito, Brygida C Bisikirska, Mariano J Alvarez, Wei Keat Lim, Presha Rajbhandari, Qiong Shen, Ilya Nemenman, Katia Basso, Adam A Margolin, Ulf Klein, Riccardo Dalla-Favera, Andrea Califano, Genome-wide identification of post-transfaltional modulators of transcription factor activity in human B cells, 2009
21. Weijun Luo, Kurt D. Hankenson, Peter J. Woolf, Learning transcriptional regulatory networks from high throughput gene expression data using continuous three-way mutual information, 2008
22. Kalpana Raja, Suresh Subramani, Jeyakumar Natarajan, A hybrid named entity recognition for tagging human proteins/genes, 2013
23. Suresh S, Kalpana R, Jeyakumar N, ProNormz – an automated web server for human proteins and protein kinases normalizations, 2011
24. Tomoko Ohta, Sampo Pyysalo, Rafal Rak, Andrew Rowley, Hong-Woo Chun, Sung-Jae Jung, Chang-Hoo Jeong Sung-Pil Choi, Jun'ichi Tsujii, Sophia Ananiadou, Overview of the pathway curation (PC) task of BioNLP Shared Task 2013, 2013
25. Alexander Nikitin, Sergei Egorov, Nikolai Daraselia, Ilya Mazo, Pathway studio – the analysis and navigation of molecular networks, 2003
26. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T, Cytoscape: a software environment for integrated models of biomolecular interaction networks, 2003

Annexes

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Generated: TCDB_families.txt -->
<!-- Generated: 2014-06-19 -->
<Ontology xmlns="exa:com.exalead.mot.components.ontology" matchOnSeparators="true">
  <Pkg path="regulation">
    <Entry display="activation">
      <Form value="activation" level="exact"/>
      <Form value="activates" level="exact"/>
      <Form value="activated" level="exact"/>
      <Form value="activate" level="exact"/>
      <Form value="activating" level="exact"/>
      <Form value="positively regulates" level="exact"/>
      <Form value="positively regulate" level="exact"/>
      <Form value="positively regulated" level="exact"/>
      <Form value="positively regulating" level="exact"/>
      <Form value="upregulated" level="exact"/>
      <Form value="upregulates" level="exact"/>
      <Form value="upregulating" level="exact"/>
    </Entry>
    <Entry display="inhibition">
      <Form value="inhibition" level="exact"/>
      <Form value="inhibits" level="exact"/>
      <Form value="inhibited" level="exact"/>
      <Form value="inhibit" level="exact"/>
      <Form value="inhibiting" level="exact"/>
      <Form value="negatively regulates" level="exact"/>
      <Form value="negatively regulate" level="exact"/>
      <Form value="negatively regulated" level="exact"/>
      <Form value="negatively regulating" level="exact"/>
      <Form value="represses" level="exact"/>
      <Form value="repressed" level="exact"/>
      <Form value="repressing" level="exact"/>
      <Form value="silences" level="exact"/>
      <Form value="silenced" level="exact"/>
      <Form value="silencing" level="exact"/>
      <Form value="downregulates" level="exact"/>
      <Form value="downregulated" level="exact"/>
      <Form value="downregulating" level="exact"/>
    </Entry>
  </Pkg>
</Ontology>
```

Annexe 1. Ontologie de terme de relation Version 0

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Generated: TCDB_families.txt -->
<!-- Generated: 2014-06-19 -->
<Ontology xmlns="exa:com.exalead.mot.components.ontology" matchOnSeparators="true">
  <Pkg path="human_context">
    <Entry display="human">
      <Form value="human" level="lemmasingularmasculine"/>
      <Form value="homo sapiens" level="lemmasingularmasculine"/>
      <Form value="patient" level="lemmasingularmasculine"/>
      <Form value="man" level="lemmasingularmasculine"/>
      <Form value="woman" level="lemmasingularmasculine"/>
      <Form value="people" level="lemmasingularmasculine"/>
      <Form value="student" level="lemmasingularmasculine"/>
      <Form value="child" level="lemmasingularmasculine"/>
      <Form value="caucasian" level="lemmasingularmasculine"/>
      <Form value="african" level="lemmasingularmasculine"/>
      <Form value="american" level="lemmasingularmasculine"/>
      <Form value="asian" level="lemmasingularmasculine"/>
      <Form value="african" level="lemmasingularmasculine"/>
      <Form value="arab" level="lemmasingularmasculine"/>
    </Entry>
  </Pkg>
</Ontology>
```

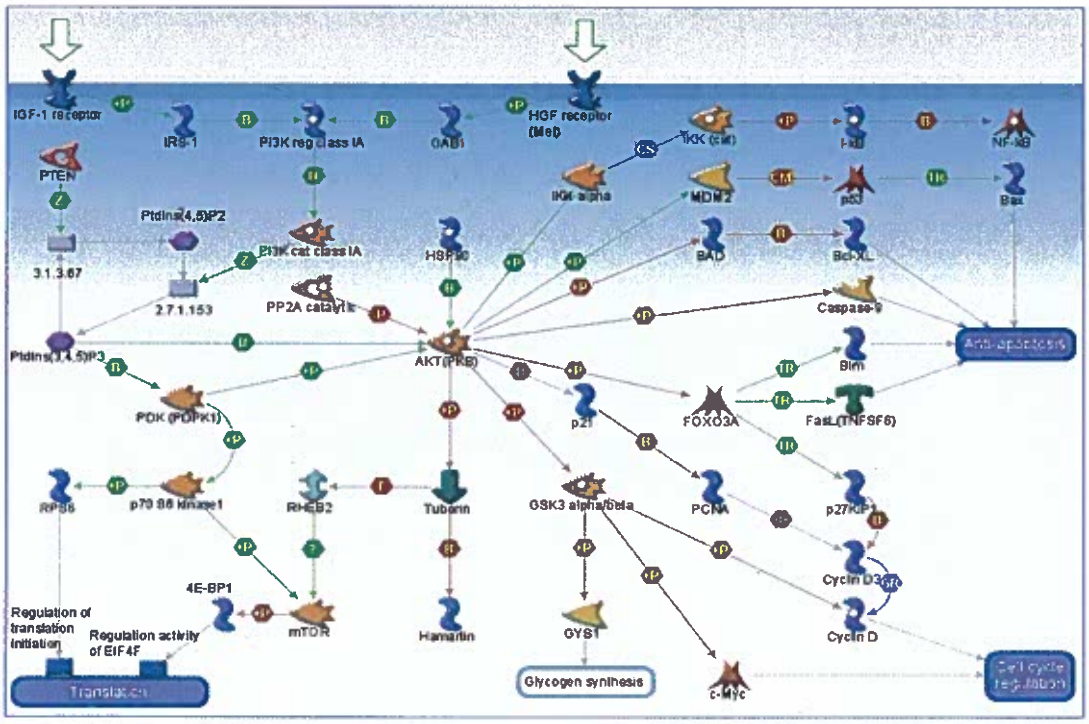
Annexe 2. Ontologie de termes humains

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Generated: TCDB_families.txt -->
<!-- Generated: 2014-06-19 -->
<Ontology xmlns="exa:com.exalead.mot.components.ontology" matchOnSeparators="true">
  <Pkg path="regulation">
    <Entry display="Abolish">
      <Form value="abolish" level="lowercase"/>
      <Form value="abolishes" level="lowercase"/>
      <Form value="abolished" level="lowercase"/>
      <Form value="abolishing" level="lowercase"/>
    </Entry>
    <Entry display="Accelerate">
      <Form value="accelerate" level="lowercase"/>
      <Form value="accelerates" level="lowercase"/>
      <Form value="accelerated" level="lowercase"/>
      <Form value="accelerating" level="lowercase"/>
    </Entry>
    <Entry display="Acceptor">
      <Form value="acceptor" level="lowercase"/>
    </Entry>
    <Entry display="Accumulate">
      <Form value="accumulate" level="lowercase"/>
      <Form value="accumulates" level="lowercase"/>
      <Form value="accumulated" level="lowercase"/>
      <Form value="accumulating" level="lowercase"/>
      <Form value="accumulation" level="lowercase"/>
    </Entry>
    <Entry display="Acetylate">
      <Form value="acetylate" level="lowercase"/>
      <Form value="acetylates" level="lowercase"/>
      <Form value="acetylated" level="lowercase"/>
      <Form value="acetylating" level="lowercase"/>
      <Form value="acetylation" level="lowercase"/>
    </Entry>
    <Entry display="Activate">
      <Form value="positively regulates" level="lowercase"/>
      <Form value="positively regulated" level="lowercase"/>
      <Form value="positively regulating" level="lowercase"/>
      <Form value="positively regulate" level="lowercase"/>
      <Form value="activate" level="lowercase"/>
      <Form value="activates" level="lowercase"/>
      <Form value="activated" level="lowercase"/>
      <Form value="activating" level="lowercase"/>
      <Form value="activation" level="lowercase"/>
      <Form value="activator" level="lowercase"/>
    </Entry>
    </Entry>
    <Entry display="Tether">
      <Form value="Tether" level="lowercase"/>
      <Form value="tethers" level="lowercase"/>
      <Form value="tethered" level="lowercase"/>
      <Form value="tethering" level="lowercase"/>
    </Entry>
    <Entry display="Transactivate">
      <Form value="transactivate" level="lowercase"/>
      <Form value="transactivates" level="lowercase"/>
      <Form value="transactivated" level="lowercase"/>
      <Form value="transactivating" level="lowercase"/>
      <Form value="transactivation" level="lowercase"/>
      <Form value="transactivator" level="lowercase"/>
    </Entry>
    <Entry display="Transaminate">
      <Form value="transaminate" level="lowercase"/>
      <Form value="transaminates" level="lowercase"/>
      <Form value="transaminated" level="lowercase"/>
      <Form value="transaminating" level="lowercase"/>
      <Form value="transamination" level="lowercase"/>
    </Entry>
    <Entry display="Ubiquitinate">
      <Form value="ubiquitinate" level="lowercase"/>
      <Form value="ubiquitinates" level="lowercase"/>
      <Form value="ubiquitinated" level="lowercase"/>
      <Form value="ubiquitinating" level="lowercase"/>
      <Form value="ubiquitination" level="lowercase"/>
    </Entry>
    <Entry display="Upregulate">
      <Form value="upregulate" level="lowercase"/>
      <Form value="upregulates" level="lowercase"/>
      <Form value="upregulated" level="lowercase"/>
      <Form value="upregulating" level="lowercase"/>
      <Form value="upregulation" level="lowercase"/>
      <Form value="upregulator" level="lowercase"/>
    </Entry>
    <Entry display="Up-regulate">
      <Form value="up-regulate" level="lowercase"/>
      <Form value="up-regulates" level="lowercase"/>
      <Form value="up-regulated" level="lowercase"/>
      <Form value="up-regulating" level="lowercase"/>
      <Form value="up-regulation" level="lowercase"/>
      <Form value="up-regulator" level="lowercase"/>
    </Entry>
  </Pkg>
</Ontology>
```

Annexe 3. Extrait de l'ontologie de termes de relation Version 1, récupéré dans les data supplementary d'un article [ ]

```
<?xml version="1.0" encoding="UTF-8"?>
<!--CloudView Ontology generated on Mon Mar 30 16:18:21 CEST 2015-->
<!--3DS UUID 6a7fd4e6-5133-4522-ad56-36e1de7e8a9c-->
<!--NCBI Gene original file /home/data/biocontent/data/test//ncbi/gene/2015_03_30/Homo_sapiens.gene_info-->
<!--UniProt original file /home/data/biocontent/data/test//uniprot/2015_03/uniprot_sprot.xml release /home/data/biocontent/data/test//uniprot/2015_03/reldate.txt-->
<Ontology matchOnSeparators="true" xmlns="exa:com.exalead.mot.components.ontology">
  <Pkg path="3ds">
    <Entry display="6a7fd4e6-5133-4522-ad56-36e1de7e8a9c">
      <Form value="6a7fd4e6-5133-4522-ad56-36e1de7e8a9c" level="lemmasingularmasculine"/>
    </Entry>
  </Pkg>
  <Pkg path="substance">
    <Entry display="gene/NCBIGene:1">
      <Form value="A1BG" level="exact"/>
      <Form value="alpha-1-B glycoprotein" level="lemmasingularmasculine"/>
      <Form value="A1B" level="exact"/>
      <Form value="ABG" level="exact"/>
      <Form value="GAB" level="exact"/>
      <Form value="HYST2477" level="exact"/>
    </Entry>
    <Entry display="gene/NCBIGene:2">
      <Form value="A2M" level="exact"/>
      <Form value="alpha-2-macroglobulin" level="lemmasingularmasculine"/>
      <Form value="A2MD" level="exact"/>
      <Form value="CPAMD5" level="exact"/>
      <Form value="FWP007" level="exact"/>
      <Form value="S863-7" level="lemmasingularmasculine"/>
    </Entry>
    <Entry display="gene/NCBIGene:3">
      <Form value="A2MP1" level="exact"/>
      <Form value="alpha-2-macroglobulin pseudogene 1" level="lemmasingularmasculine"/>
      <Form value="A2MP" level="exact"/>
    </Entry>
    <Entry display="gene/NCBIGene:9">
      <Form value="NAT1" level="exact"/>
      <Form value="N-acetyltransferase 1 (arylamine N-acetyltransferase)" level="lemmasingularmasculine"/>
      <Form value="AAC1" level="exact"/>
      <Form value="MNAT" level="exact"/>
      <Form value="NAT-1" level="lemmasingularmasculine"/>
      <Form value="NATI" level="exact"/>
    </Entry>
```

Annexe 4. Extrait de l'ontologie substance, contenant des millions de gènes et protéines



Annexe 5. MAP GeneGo provenant du Free Pathway 'Signal transduction_AKT signaling'

References:

1. Katso R, Okkenhaug K, Ahmadi K, White S, Timms J, Waterfield MD
Cellular function of phosphoinositide 3-kinases: implications for development, homeostasis, and cancer. Annual review of cell and developmental biology 2001;17:615-75

2. Brader S, Eccles SA
Phosphoinositide 3-kinase signalling pathways in tumor progression, invasion and angiogenesis. Tumori 2004 Jan-Feb;90(1):2-8

3. Downward J
PI 3-kinase, Akt and cell survival. Seminars in cell & developmental biology 2004 Apr;15(2):177-82

4. Sato S, Fujita N, Tsuruo T
Modulation of Akt kinase activity by binding to Hsp90. Proceedings of the National Academy of Sciences of the United States of America 2000 Sep 26;97(20):10832-7

5. Tsugawa K, Jones MK, Sugimachi K, Sarfeh IJ, Tarnawski AS
Biological role of phosphatase PTEN in cancer and tissue injury healing. Frontiers in bioscience : a journal and virtual library 2002 May 1;7:e245-51

6. Datta SR, Ranger AM, Lin MZ, Sturgill JF, Ma YC, Cowan CW, Dikkes P, Korsmeyer SJ, Greenberg ME
Survival factor-mediated BAD phosphorylation raises the mitochondrial threshold for apoptosis. Developmental cell 2002 Nov;3(5):631-43

7. Kau TR, Schroeder F, Ramaswamy S, Wojciechowski CL, Zhao JJ, Roberts TM, Clardy J, Sellers WR, Silver PA
A chemical genetic screen identifies inhibitors of regulated nuclear export of a Forkhead transcription factor in PTEN-deficient tumor cells. Cancer cell 2003 Dec;4(6):463-76

8. Agarwal A, Das K, Lerner N, Sathe S, Cicek M, Casey G, Sizemore N
The AKT/I kappa B kinase pathway promotes angiogenic/metastatic gene expression in colorectal cancer by activating nuclear factor-kappa B and beta-catenin. Oncogene 2005 Feb 3;24(6):1021-31

9. Shen Y, White E
p53-dependent apoptosis pathways. Advances in cancer research 2001;82:55-84

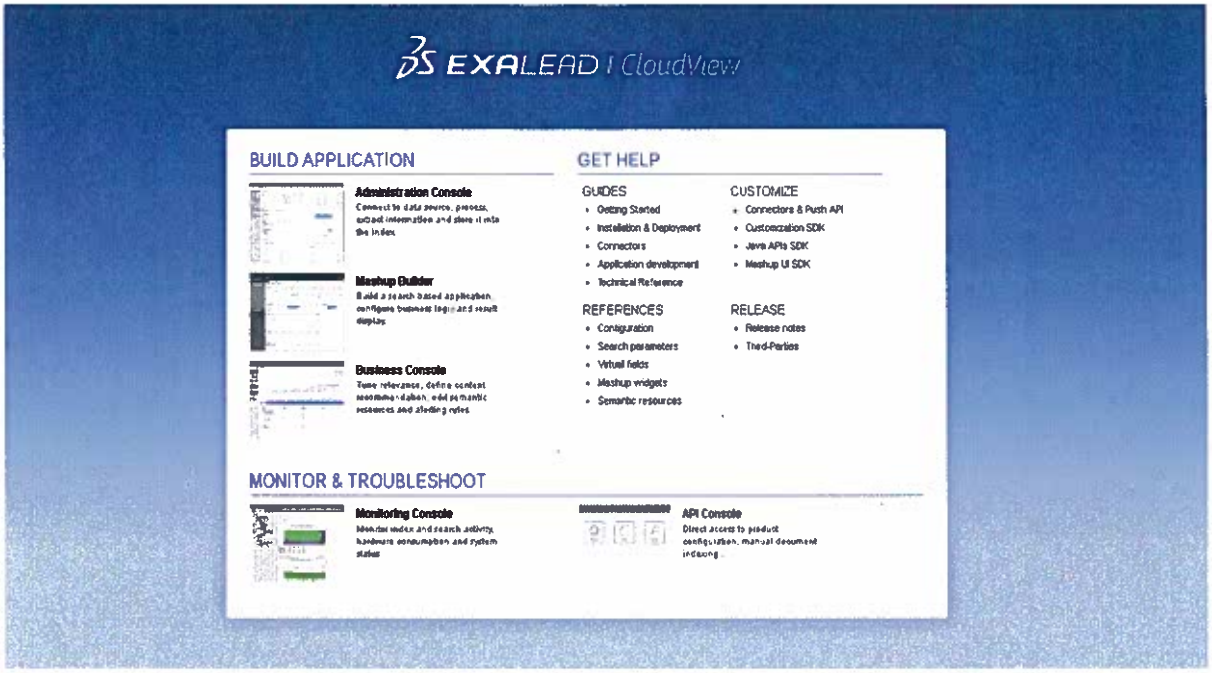
10. Gottlieb TM, Leal JF, Seger R, Taya Y, Oren M
Cross-talk between Akt, p53 and Mdm2: possible implications for the regulation of apoptosis. Oncogene 2002 Feb 14;21(8):1299-303

11. Zhou BP, Liao Y, Xia W, Spohn B, Lee MH, Hung MC
Cytoplasmic localization of p21Cip1/WAF1 by Akt-induced phosphorylation in HER-2/neu-overexpressing cells. Nature cell biology 2001 Mar;3(3):245-52

12. Liang J, Slingerland JM
Multiple roles of the PI3K/PKB (Akt) pathway in cell cycle progression. Cell cycle (Georgetown, Tex.) 2003 Jul-Aug;2(4):339-45

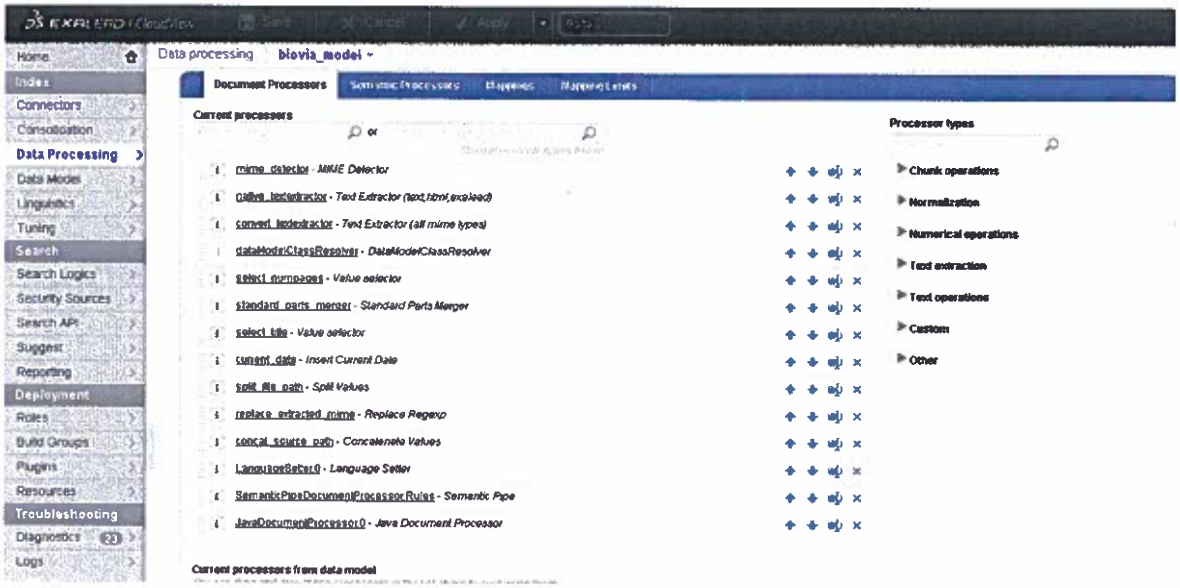
13. Panwalkar A, Verstovsek S, Giles FJ
Mammalian target of rapamycin inhibition as therapy for hematologic malignancies. Cancer 2004 Feb 15;100(4):657-66

Annexe 6. Liste des articles associées au Free Pathway de l'annexe 5



Annexe 7. Capture écran de l'interface admin CloudView

Annexe 8. Capture écran de la console d'administration de CloudView, où apparaissent la liste des connecteurs



Annexe 9. Capture écran du pipeline d'analyse

Annexe 10. Capture écran du pipeline sémantique

	A	B	C	D	E	F
1	PATTERN	INTERACTOR	RELATION	INTERACTOR2	INTACT SCORE	PATHS
2	uniprot:P00533 TKIs, inhibiting uniprot:P41240	uniprot:P00533	inhibiting	uniprot:P41240	2	32504
3	uniprot:P01135) may modulate uniprot:P00533	uniprot:P01135	modulate	uniprot:P00533	2	3469
4	uniprot:Q99962 (associated with uniprot:P00533	uniprot:Q99962	associated	uniprot:P00533	2	40945
5	uniprot:P00533 signaling pathways induce cancer cell uniprot:Q08881	uniprot:P00533	induce	uniprot:Q08881	2	21356
6	Induction of uniprot:P00533 ligands and utilizes EGFR signaling to increase IL-8 and uniprot:F	uniprot:P00533	induction	uniprot:P05362	2	1735
7	uniprot:P03372 (ER) ? activated by 17?-estradiol (E?) increased uniprot:P00533	uniprot:P03372	activated	uniprot:P00533	2	493779
8	uniprot:Q62245 protein associated with the uniprot:P00533	uniprot:Q62245	associated	uniprot:P00533	2	2959
9	uniprot:P00533, lysophosphatidic acid-induced uniprot:P98083	uniprot:P00533	induced	uniprot:P98083	2	7137
10	suppression by uniprot:P00533 activation and immune reactivation by EGFR-TKIs and/or unip	uniprot:P00533	suppression	uniprot:P18621	2	15966
11	Substitution of six basic amino acid residues within the CaM-binding domain (uniprot:P6220	uniprot:P62204	Substitution	uniprot:P00533	2	333
12	phosphorylation of c-RAF and PI3K uniprot:Q14155, upstream of MAPK and downstream of u	uniprot:Q14155	phosphorylation	uniprot:P00533	2	10305
13	uniprot:P35568 was phosphorylated on tyrosines upon incubation with purified uniprot:P00	uniprot:P35568	phosphorylated	uniprot:P00533	2	20708
14	suppression of uniprot:P17936 did not affect sensitivity to uniprot:P00533	uniprot:P17936	suppression	uniprot:P00533	2	7331
15	uniprot:P00533-induced cell migration and indicate that targeting of uniprot:Q9H5V8	uniprot:P00533	induced	uniprot:Q9H5V8	2	7426
16	uniprot:Q38SD2 regulates uniprot:P00533	uniprot:Q38SD2	regulates	uniprot:P00533	2	108246
17	Inactivation of uniprot:P00533 inhibits the anchorage independence and AKT pathway prom	uniprot:P00533	inactivation	uniprot:P16144	2	9858
18	uniprot:P00533 signaling in regulating host defense and immune response by tightly control	uniprot:P00533	regulating	uniprot:O60603	2	15955
19	uniprot:P18031 activated uniprot:P00533	uniprot:P18031	activated	uniprot:P00533	2	97118
20	uniprot:P26447 stimulated uniprot:P00533	uniprot:P26447	stimulated	uniprot:P00533	2	3744
21	uniprot:O75165 In sorting decisions influencing uniprot:P00533	uniprot:O75165	influencing	uniprot:P00533	2	7951
22	express EphB4, which is induced either by uniprot:P00533 activation or by uniprot:P54760	uniprot:P00533	express	uniprot:P54760	2	1813
23	uniprot:P00533 and its adaptors accumulate in uniprot:Q15075	uniprot:P00533	accumulate	uniprot:Q15075	2	22968
24	induction of CTGF expression by uniprot:Q02297, as did treating cells with the uniprot:P0053	uniprot:Q02297	induction	uniprot:P00533	2	1357
25	uniprot:P00533 enhances uniprot:P33993	uniprot:P00533	enhances	uniprot:P33993	2	256740
26	amplification of the uniprot:P00533 ErbB2 in mammary tissue, correlates with a marked up-r	uniprot:P00533	amplification	uniprot:Q01469	2	12387
27	overexpression contributes to uniprot:P00533-TKI resistance in NSCLC and that uniprot:P051	uniprot:P00533	overexpression	uniprot:P05141	2	42269
28	association of uniprot:Q9Y2R2 rs2476601 as well as epidermal growth factor receptor (uniprc	uniprot:Q9Y2R2	association	uniprot:P00533	2	14744
29	uniprot:P00533-induced cleavage could be shown to lead to degradation of the catalytic unip	uniprot:P00533	induced	uniprot:P10586	2	5916
30	activation and trafficking of uniprot:P00533 (EGFR) induced by UV light and uniprot:P01133	uniprot:P00533	activation	uniprot:P01133	14	2394
31	expression levels of uniprot:P00533 and HER2 were treated with specific EGFR and bispecific	uniprot:P00533	expression	uniprot:P04626	8	601512
32	upregulation of uniprot:Q9UBN7 might act to slow the trafficking of uniprot:P00533	uniprot:Q9UBN7	upregulation	uniprot:P00533	7	117021
33	uniprot:P20936) that inhibits the trafficking of uniprot:P00533	uniprot:P20936	inhibits	uniprot:P00533	3	78564
34	uniprot:P22682 proteins regulate the endocytic trafficking of the uniprot:P00533	uniprot:P22682	regulate	uniprot:P00533	3	351
35	uniprot:Q96B97 regulates endocytic trafficking of the uniprot:P00533	uniprot:Q96B97	regulates	uniprot:P00533	4	12525
36	uniprot:P00533, and expressing uniprot:Q13480	uniprot:P00533	expressing	uniprot:Q13480	3	159267
37	production of EGFR ligands and mediates radioresistance through uniprot:P00533-depender	uniprot:P00533	production	uniprot:P97313	3	705
38	phosphorylation of ERBB2 or MET were associated with reduced sensitivity to acute loss of u	uniprot:P21860	phosphorylation	uniprot:P00533	7	163435

Annexe 11. Fichier CSV des patterns extraits pour EGFR

	A	B	C	D
1	INTERACTOR 1	INTERACTOR 2	NB INTACT	NB DE PATTERN
2	uniprot:P00533	uniprot:Q13882	1	2
3	uniprot:P00533	uniprot:O60674	1	2
4	uniprot:P00533	uniprot:P01135	1	8
5	uniprot:P00533	uniprot:O60603	1	2
6	uniprot:P00533	uniprot:Q60631	1	32
7	uniprot:P00533	uniprot:Q12913	1	4
8	uniprot:P00533	uniprot:P98083	1	3
9	uniprot:P00533	uniprot:P50281	1	6
10	uniprot:P00533	uniprot:P35222	1	2
11	uniprot:P00533	uniprot:P07949	1	1
12	uniprot:P00533	uniprot:O14543	1	2
13	uniprot:P00533	uniprot:P17936	1	4
14	uniprot:P00533	uniprot:Q08881	1	19
15	uniprot:P00533	uniprot:P19438	1	4
16	uniprot:P00533	uniprot:P15311	1	4
17	uniprot:P00533	uniprot:Q05397	1	1
18	uniprot:P00533	uniprot:Q07065	1	2
19	uniprot:P00533	uniprot:P00533	23	774
20	uniprot:P00533	uniprot:P20936	5	6
21	uniprot:P00533	uniprot:P62993	26	2
22	uniprot:P00533	uniprot:P04626	14	21
23	uniprot:P00533	uniprot:Q15303	2	1
24	uniprot:P00533	uniprot:Q06124	2	10
25	uniprot:P00533	uniprot:P15941	4	12
26	uniprot:P00533	uniprot:P01133	13	274
27	uniprot:P00533	uniprot:Q03135	5	6
28	uniprot:P00533	uniprot:P62158	2	15
29	uniprot:P00533	uniprot:P07948	4	1
30	uniprot:P00533	uniprot:P03372	2	8
31	uniprot:P00533	uniprot:P13866	3	2
32	uniprot:P00533	uniprot:P18031	6	4
33	uniprot:P00533	uniprot:P40763	9	6
34	uniprot:P00533	uniprot:P42224	4	4
35	uniprot:P00533	uniprot:P04083	4	1
36	uniprot:P00533	uniprot:P46940	4	4
37	uniprot:P00533	uniprot:P12830	3	8
38	uniprot:P00533	uniprot:Q00325	3	5
39	uniprot:P00533	uniprot:P43307	2	1
40	uniprot:P00533	uniprot:P97313	2	4

Annexe 12. Fichier CSV des interactions Intact avec le nombre de patterns contenant les mêmes interactions