



# Suggestion d'interactions gène-médicament à partir de données ouvertes et liées

Kevin Dalleau

## ► To cite this version:

Kevin Dalleau. Suggestion d'interactions gène-médicament à partir de données ouvertes et liées. Sciences pharmaceutiques. 2017. hal-01945028

**HAL Id: hal-01945028**

**<https://hal.univ-lorraine.fr/hal-01945028>**

Submitted on 5 Dec 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



## AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : [ddoc-thesesexercice-contact@univ-lorraine.fr](mailto:ddoc-thesesexercice-contact@univ-lorraine.fr)

## LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

[http://www.cfcopies.com/V2/leg/leg\\_droi.php](http://www.cfcopies.com/V2/leg/leg_droi.php)

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

**UNIVERSITE DE LORRAINE**  
**2017**

---

**FACULTE DE PHARMACIE**

**T H E S E**

Présentée et soutenue publiquement

le 12 Juillet 2017, sur un sujet dédié à :

**SUGGESTION D'INTERACTIONS GÈNE-MÉDICAMENT  
À PARTIR DE DONNÉES OUVERTES  
ET LIÉES**

pour obtenir

**le Diplôme d'Etat de Docteur en Pharmacie**

par  
**Kevin DALLEAU**  
né le 08 Avril 1989

**Membres du Jury**

Président : Monsieur Luc Ferrari, Professeur de Universités

Juges :   Monsieur Adrien Coulet, Maître de Conférences  
          Madame Ndeye-Coumba Ndiaye, Maître de Conférences  
          Madame Nadine Petitpain, Praticien Hospitalier

**UNIVERSITÉ DE LORRAINE  
FACULTÉ DE PHARMACIE  
Année universitaire 2016-2017**

**DOYEN**

Francine PAULUS

**Vice-Doyen**

Béatrice FAIVRE

**Directeur des Etudes**

Virginie PICHON

**Conseil de la Pédagogie**

Président, Brigitte LEININGER-MULLER

**Collège d'Enseignement Pharmaceutique Hospitalier**

Président, Béatrice DEMORE

**Commission Prospective Facultaire**

Président, Christophe GANTZER

Vice-Président, Jean-Louis MERLIN

**Commission de la Recherche**

Président, Raphaël DUVAL

**Responsable de la filière Officine**

**Responsables de la filière Industrie**

**Responsable de la filière Hôpital**

**Responsable Pharma Plus ENSIC**

**Responsable Pharma Plus ENSAIA**

**Responsable Pharma Plus ENSGSI**

**Responsable de la Communication**

**Responsable de la Cellule de Formation Continue  
et individuelle**

**Responsable de la Commission d'agrément  
des maîtres de stage**

**Responsable ERASMUS**

Béatrice FAIVRE

Isabelle LARTAUD,

Jean-Bernard REGNOUF de VAINS

Béatrice DEMORE

Jean-Bernard REGNOUF de VAINS

Raphaël DUVAL

Igor CLAROT

Marie-Paule SAUDER

Béatrice FAIVRE

Béatrice FAIVRE

Mihayl VARBANOV

**DOYENS HONORAIRES**

Chantal FINANCE

Claude VIGNERON

**PROFESSEURS EMERITES**

Jeffrey ATKINSON

Jean-Claude BLOCK

Max HENRY

Alain MARSURA

Claude VIGNERON

**PROFESSEURS HONORAIRES**

Roger BONALY

Pierre DIXNEUF

Marie-Madeleine GALTEAU

Thérèse GIRARD

Michel JACQUE

Pierre LABRUDE

Vincent LOPPINET

Janine SCHWARTZBROD

Louis SCHWARTZBROD

**MAITRES DE CONFERENCES HONORAIRES**

Monique ALBERT

Mariette BEAUD

Gérald CATAU

Jean-Claude CHEVIN

Jocelyne COLLOMB

Bernard DANGIEN

Marie-Claude FUZELLIER

Françoise HINZELIN

Francine KEDZIEREWICZ

Marie-Hélène LIVERTOUX

**ASSISTANTS HONORAIRES**

Marie-Catherine BERTHE  
Annie PAVIS

Bernard MIGNOT  
Jean-Louis MONAL  
Blandine MOREAU  
Dominique NOTTER  
Christine PERDICAKIS  
Marie-France POCHON  
Anne ROVEL  
Gabriel TROCKLE  
Maria WELLMAN-ROUSSEAU  
Colette ZINUTTI

**ENSEIGNANTS**

Section CNU\*

Discipline d'enseignement

**PROFESSEURS DES UNIVERSITES - PRATICIENS HOSPITALIERS**

Danièle BENSOUSSAN-LEJZEROWICZ	82	Thérapie cellulaire
Jean-Louis MERLIN	82	Biologie cellulaire
Alain NICOLAS	80	Chimie analytique et Bromatologie
Jean-Michel SIMON	81	Economie de la santé, Législation pharmaceutique
Nathalie THILLY	81	Santé publique et Epidémiologie

**PROFESSEURS DES UNIVERSITES**

Christine CAPDEVILLE-ATKINSON	86	Pharmacologie
Igor CLAROT ☞	85	Chimie analytique
Joël DUCOURNEAU	85	Biophysique, Acoustique, Audioprothèse
Raphaël DUVAL	87	Microbiologie clinique
Béatrice FAIVRE	87	Biologie cellulaire, Hématologie
Luc FERRARI	86	Toxicologie
Pascale FRIANT-MICHEL	85	Mathématiques, Physique
Christophe GANTZER	87	Microbiologie
Frédéric JORAND	87	Eau, Santé, Environnement
Isabelle LARTAUD	86	Pharmacologie
Dominique LAURAIN-MATTAR	86	Pharmacognosie
Brigitte LEININGER-MULLER	87	Biochimie
Pierre LEROY	85	Chimie physique
Philippe MAINCENT	85	Pharmacie galénique
Patrick MENU	86	Physiologie
Jean-Bernard REGNOUF de VAINS	86	Chimie thérapeutique
Bertrand RIHN	87	Biochimie, Biologie moléculaire

**MAITRES DE CONFÉRENCES DES UNIVERSITÉS - PRATICIENS HOSPITALIERS**

Béatrice DEMORE	81	Pharmacie clinique
Alexandre HARLE ☞	82	Biologie cellulaire oncologique
Julien PERRIN	82	Hématologie biologique
Marie SOCHA	81	Pharmacie clinique, thérapeutique et biotechnique

**MAITRES DE CONFÉRENCES**

Sandrine BANAS	87	Parasitologie
Xavier BELLANGER	87	Parasitologie, Mycologie médicale
Emmanuelle BENOIT	86	Communication et Santé
Isabelle BERTRAND	87	Microbiologie
Michel BOISBRUN	86	Chimie thérapeutique
François BONNEAUX	86	Chimie thérapeutique

Ariane BOUDIER	85	Chimie Physique
Cédric BOURA	86	Physiologie
Joël COULON	87	Biochimie
Sébastien DADE	85	Bio-informatique
Dominique DECOLIN	85	Chimie analytique
Roudayna DIAB	85	Pharmacie galénique
Natacha DREUMONT	87	Biochimie générale, Biochimie clinique
Florence DUMARCAY	86	Chimie thérapeutique
François DUPUIS	86	Pharmacologie
Adil FAIZ	85	Biophysique, Acoustique
Anthony GANDIN	87	Mycologie, Botanique
Caroline GAUCHER	86	Chimie physique, Pharmacologie
Stéphane GIBAUD	86	Pharmacie clinique
Thierry HUMBERT	86	Chimie organique
Olivier JOUBERT	86	Toxicologie, Sécurité sanitaire
<b>ENSEIGNANTS (suite)</b>	<b>Section CNU*</b>	<b>Discipline d'enseignement</b>
Alexandrine LAMBERT	85	Informatique, Biostatistiques
Julie LEONHARD	86/01	Droit en Santé
Christophe MERLIN	87	Microbiologie environnementale
Maxime MOURER	86	Chimie organique
Coumba NDIAYE	86	Epidémiologie et Santé publique
Marianne PARENT ☞	85	Pharmacie galénique
Francine PAULUS	85	Informatique
Caroline PERRIN-SARRADO	86	Pharmacologie
Virginie PICHON	85	Biophysique
Sophie PINEL	85	Informatique en Santé (e-santé)
Anne SAPIN-MINET	85	Pharmacie galénique
Marie-Paule SAUDER	87	Mycologie, Botanique
Guillaume SAUTREY	85	Chimie analytique
Rosella SPINA	86	Pharmacognosie
Sabrina TOUCHET ☞	86	Pharmacochimie
Mihayl VARBANOV	87	Immuno-Virologie
Marie-Noëlle VAULTIER	87	Mycologie, Botanique
Emilie VELOT	86	Physiologie-Physiopathologie humaines
Mohamed ZAIOU	87	Biochimie et Biologie moléculaire
<b>PROFESSEUR ASSOCIE</b>		
Anne MAHEUT-BOSSER	86	Sémiologie
<b>PROFESSEUR AGREGÉ</b>		
Christophe COCHAUD	11	Anglais

☞ En attente de nomination

*\*Disciplines du Conseil National des Universités :*

80 : Personnels enseignants et hospitaliers de pharmacie en sciences physico-chimiques et ingénierie appliquée à la santé

81 : Personnels enseignants et hospitaliers de pharmacie en sciences du médicament et des autres produits de santé

82 : Personnels enseignants et hospitaliers de pharmacie en sciences biologiques, fondamentales et cliniques

85 : Personnels enseignants-chercheurs de pharmacie en sciences physico-chimiques et ingénierie appliquée à la santé

86 : Personnels enseignants-chercheurs de pharmacie en sciences du médicament et des autres produits de santé

87 : Personnels enseignants-chercheurs de pharmacie en sciences biologiques, fondamentales et cliniques

11 : Professeur agrégé de lettres et sciences humaines en langues et littératures anglaises et anglo-saxonnes

## **SERMENT DES APOTHICAIRES**

**Je jure, en présence des maîtres de la Faculté, des conseillers de l'ordre des pharmaciens et de mes condisciples :**

**D'honorer ceux qui m'ont instruit dans les préceptes de mon art et de leur témoigner ma reconnaissance en restant fidèle à leur enseignement.**

**D'exercer, dans l'intérêt de la santé publique, ma profession avec conscience et de respecter non seulement la législation en vigueur, mais aussi les règles de l'honneur, de la probité et du désintéressement.**

**De ne jamais oublier ma responsabilité et mes devoirs envers le malade et sa dignité humaine ; en aucun cas, je ne consentirai à utiliser mes connaissances et mon état pour corrompre les mœurs et favoriser des actes criminels.**

**Que les hommes m'accordent leur estime si je suis fidèle à mes promesses.**

**Que je sois couvert d'opprobre et méprisé de mes confrères si j'y manque.**

« LA FACULTE N'ENTEND DONNER AUCUNE APPROBATION, NI IMPROBATION  
AUX OPINIONS EMISES DANS LES THESES, CES OPINIONS DOIVENT ETRE  
CONSIDEREES COMME PROPRES A LEUR AUTEUR ».



## **Remerciements**

### **A Monsieur Adrien Coulet**

Merci d'avoir accepté de diriger cette thèse. Plus généralement, je tenais à te remercier pour ton encadrement, tes conseils, ta disponibilité et ton aide, sans lesquels je n'aurais peut-être pas été là où je suis actuellement.

### **A Madame Coumba-Ndeye Ndiaye**

Merci d'avoir accepté de diriger cette thèse, et plus généralement, pour tous vos conseils et pour votre disponibilité. Je tiens à vous exprimer ma sincère reconnaissance.

### **A Monsieur Luc Ferrari**

Merci d'avoir accepté de présider ce jury, et pour votre grande disponibilité. Que cette thèse soit l'expression de toute ma reconnaissance à votre égard.

### **A Madame Nadine Petitpain**

Merci d'avoir accepté de juger cette thèse. Soyez assurée de mon respect et ma gratitude.

### **A mes parents, mes grands parents, mon parrain**

Sans vous, rien de tout ça n'aurait été possible. Merci pour tout, merci pour votre soutien sans faille pendant toutes ces années !

### **A tous mes proches**

À Mylène et Sophie : merci si la famille !

À vous qui avez fait de ces années d'études des années inoubliables : Floriane, Nastasia, Pierre, Rahim, Patricia, Alex, Jean-Philippe, Sylvain, Quentin... Vous qui faites partie de cette famille que l'on choisit : merci !

### **Et à tous ceux qui ont rendu ce travail possible**

Yassine, pour ton travail sur les noyaux. Un grand merci, et bonne continuation !  
Patrice et Sébastien, pour vos contributions respectives à l'article, merci.  
Malika, pour m'avoir laissé la chance de continuer cette aventure dans le monde de la recherche, et pour ton encadrement.

# Table des matières

<b>I</b>	<b>Introduction générale</b>	<b>1</b>
<b>II</b>	<b>État de l’art et positionnement</b>	<b>3</b>
<b>1</b>	<b>La pharmacogénomique</b>	<b>3</b>
1.1	Le métabolisme et l’action du médicament . . . . .	6
1.1.1	La pharmacocinétique . . . . .	6
1.1.2	La pharmacodynamique . . . . .	8
1.1.3	Défis de la pharmacogénomique . . . . .	10
<b>2</b>	<b>L’extraction de connaissances à partir des bases de données</b>	<b>10</b>
2.1	Définition . . . . .	10
2.2	Le processus de fouille de données . . . . .	11
2.3	Les méthodes non supervisées . . . . .	12
2.3.1	La méthode des k-moyennes . . . . .	12
2.3.2	Le clustering hiérarchique . . . . .	17
2.4	Les méthodes supervisées . . . . .	20
2.4.1	Les arbres de décision . . . . .	20
2.4.2	Les forêts d’arbres aléatoires . . . . .	24
2.4.3	Les machines à vecteur de support . . . . .	25
2.5	Les méthodes de fouille appliquées aux graphes . . . . .	29
2.5.1	Les graphes et leur représentation . . . . .	29
2.5.2	Application des méthodes de fouilles aux graphes : notion de noyaux de graphes . . . . .	36
2.6	Évaluation des tâches d’apprentissage . . . . .	38
<b>3</b>	<b>Le Web des données</b>	<b>41</b>
3.1	Web des données, Web sémantique, Données ouvertes et liées : définitions . .	41
3.2	Un ensemble de standards . . . . .	42

3.2.1	<i>One entity to rule them all</i> : le W3C . . . . .	42
3.2.2	Une identité associée aux objets : les URI . . . . .	43
3.2.3	Un modèle standard de représentation de données : RDF . . . . .	45
3.2.4	Un langage de requêtes : SPARQL . . . . .	45
<b>4</b>	<b>Positionnement</b>	<b>47</b>
<b>5</b>	<b>Contributions</b>	<b>49</b>
<b>III</b>	<b>Suggestion de pharmacogènes valides par apprentissage à partir de données ouvertes et liées</b>	<b>50</b>
<b>IV</b>	<b>Discussion et conclusion</b>	<b>63</b>
	<b>Glossaire</b>	<b>75</b>

# Table des figures

1	Exemple de la variation de réponse à une molécule. . . . .	4
2	Médicaments approuvés par la FDA dont la notice contient des éléments concernant la génétique au sein de leur indication. . . . .	5
3	Processus de KDD (Knowledge Discovery in Databases). . . . .	11
4	Ensemble d'étapes de la méthode des k-moyennes. . . . .	15
5	Exemple de région convexe dans un espace à deux dimensions. . . . .	16
6	Exemple de région non convexe. . . . .	16
7	Dendrogramme représentant les différences génétiques entre 17 populations.	17
8	Représentation schématique de la distance entre deux clusters A et B, selon le critère de <i>complete linkage</i> . . . . .	18
9	Représentation schématique des distances entre deux clusters A et B, avec le critère de l' <i>average linkage</i> . . . . .	19
10	Représentation schématique de la distance entre deux clusters A et B, selon le critère du <i>single linkage</i> . . . . .	19
11	Division du jeu de données en fonction de l'attribut <i>Température</i> . . . . .	22
12	Exemple d'arbre de décision. . . . .	23
13	Exemple d'hyperplan séparateur. . . . .	25
14	Ensemble d'objets séparés par un hyperplan maximisant la marge. . . . .	27
15	Ensemble d'objets non séparables de manière linéaire. . . . .	28
16	Exemple de graphe simple. . . . .	29
17	Matrice d'adjacence du graphe simple de la Figure 16. . . . .	30
18	Matrice des degrés du graphe simple de la Figure 16. . . . .	31
19	Exemple de parcours en largeur. . . . .	33
20	Exemple de parcours en profondeur. . . . .	35
21	Etapes 1 et 2 du calcul du <i>Weisfeiler-Lehman Subtree Kernel</i> . . . . .	37
22	Etapes 3 et 4 du calcul du <i>Weisfeiler-Lehman Subtree Kernel</i> . . . . .	37
23	Dernière étape du calcul du <i>Weisfeiler-Lehman Subtree Kernel</i> . . . . .	38
24	Le nuage des données ouvertes et liées. . . . .	43

25	Zoom sur les données liées du monde médical. . . . .	44
26	Exemple de graphe RDF. . . . .	46
27	Exemple de graphe représenté au format RDF-XML. . . . .	47
28	Représentation graphique des liaisons indirectes entre médicaments passant par des gènes. . . . .	48
29	Présentation des objectifs du projet PractiKPharma. . . . .	65

Liste des tableaux

1	Jeu de données d'apprentissage. . . . .	21
2	Exemple de matrice de confusion . . . . .	39

## Première partie

# Introduction générale

La pharmacogénomique étudie les variations interindividuelles de réponse à un médicament, liées aux différences génétiques entre personnes ou populations [1]. De solides connaissances constituent une base à l'implémentation de stratégies thérapeutiques spécialisées, *i.e.*, un traitement adapté à chaque patient en considérant –entre autres–, son patrimoine génétique.

A l'heure actuelle, une grande partie des connaissances du domaine est contenue dans la littérature biomédicale, ainsi que dans des bases de données spécialisées [2, 3]. Cependant, une grande partie de ces connaissances n'est pas validée, et n'est pas encore utilisable en pratique. En effet, ces dernières proviennent d'études difficilement reproductibles, et ayant parfois une validité statistique limitée, au vu de la rareté de certains variants génétiques, ainsi que de la possibilité de coaction entre variants [4][5]. Il est par conséquent intéressant pour la communauté impliquée dans la pharmacogénomique et la médecine personnalisée d'explorer de nouvelles sources de preuves pouvant contribuer à confirmer ou modérer ces connaissances insuffisamment validées.

Les travaux précédents se focalisaient sur l'étude de bases de données spécifiques ou de la littérature biomédicale. Hansen *et al.*, par exemple, proposent dans [6] d'identifier des pharmacogènes candidats à partir de données provenant des bases de PharmGKB, DrugBank et d'interactions protéines–protéines provenant d'InWeb. Le problème de cette approche est que PharmGKB et DrugBank sont des sources de données alimentées manuellement à partir de la littérature, et sont par conséquent incomplètes. Garten *et al.* ont proposé une méthode automatisée traitant directement la littérature – et uniquement la littérature –, contournant ainsi cette problématique [7].

L'utilisation des technologies du Web sémantique a elle aussi déjà été expérimentée dans la découverte de connaissances en pharmacogénomique. Dumontier et Villanueva-Rosales

ont proposé une méthode de représentation des connaissances du domaine permettant de répondre à des requêtes complexes, dans un cadre bien spécifique, celui des traitements de la dépression [8]. Coulet *et al.* ont utilisé des données de patients dans le but de mettre en place une base de connaissances, employée afin d'extraire des règles d'association et, *in fine*, identifier des associations entre variants génétiques et réponse aux médicaments [9]. Plus généralement, les divers apports que le Web sémantique pourrait avoir pour la recherche en pharmacogénomique sont listés dans [10].

Nous présentons ici une méthode consistant à fouiller diverses sources de données représentées selon les standards des données ouvertes et liées, afin de suggérer des relations entre variants et médicaments. Dans une première partie, un état de l'art est présenté, introduisant les différents concepts clés. Puis, l'article publié dans *Journal of biomedical semantics* décrit les différentes étapes de sélection, prétraitement et fouille, ainsi que les résultats obtenus. Enfin, dans une dernière section, nous concluons et décrivons certaines perspectives.



## Deuxième partie

# État de l’art et positionnement

## 1 La pharmacogénomique

Un médicament, lors de sa mise sur le marché, possède une ou plusieurs indication(s), définie(s) dans son autorisation de mise sur le marché. L’action d’un médicament est, tout logiquement, liée à son indication. Un médicament ayant pour indication «Toux sèche» a par exemple une action antitussive. L’action d’un médicament n’est cependant pas strictement identique sur l’ensemble des individus. Cette dernière dépend de nombreux paramètres, tels que l’âge [11], la masse [12], la prise concomitante d’autres substances [13].

La variation de la réponse à un médicament n’est pas sans conséquence clinique, pouvant aller de l’absence de réponse à un traitement, à de sévères effets indésirables pouvant entraîner des séquelles graves, voire la mort. Une méta analyse réalisée en 1998 par Lazarou J. *et al.* a montré que les effets indésirables constitueraient entre la 4e et la 6e cause de décès aux États-Unis [14]. La recherche des facteurs pouvant mener à de tels effets revêt donc un intérêt majeur de santé publique.

Parmi les causes de variations de la réponse à un médicament, la génétique joue un rôle non négligeable. L’étude de l’effet des variations génétiques sur l’effet d’une substance thérapeutique constitue le cœur de la pharmacogénomique, champ de recherche qui étudie comment les variations génétiques interindividuelles impactent la réponse aux médicaments [15]. Bien que ce terme soit apparu pour la première fois au cours des années 1950 [16], ce n’est que récemment que l’intérêt porté par la communauté scientifique a grandi, avec l’amélioration des connaissances en biologie moléculaire et en biotechnologies. Ces dernières ont permis l’accélération et la diminution du coût du séquençage du génome, rendant accessibles à un plus grand nombre des stratégies thérapeutiques ciblées.

Par ailleurs, une précision terminologique est à faire : alors que la pharmacogénomique s'intéresse à l'ensemble d'un génotype, la pharmacogénétique en général, s'intéresse aux effets de la variation d'un seul gène [17]. Cependant, ces deux termes sont de plus en plus utilisés de manière interchangeable dans la littérature.

La connaissance de ces relations entre variants et réponses individuelles permettent (i) de prédire, et donc prévenir, les réactions indésirables (toxicité) ; et (ii), d'adapter individuellement les traitements.

La pharmacogénomique peut être vue comme l'étude des relations ternaires entre 3 entités : un médicament, un génotype, et un phénotype [18]. Un exemple de relation de ce type est présenté figure 1. Il présente les variations de réponse à une même substance, la codéine, en fonction de variants d'un même gène, ici *CYP2D6*.

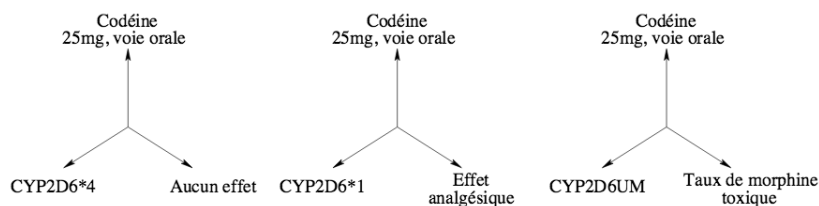


FIGURE 1 – Exemple de la variation de réponse à une molécule, la codéine [18].

Dans cet exemple, on constate que l'un des variants (*CYP2D6\*4*) est lié à l'absence complète d'effet de la molécule, alors qu'un autre (*CYP2D6UM*) modifie le métabolisme de la morphine de telle sorte que la prise de la molécule entraîne des effets toxiques. On comprend bien ici l'intérêt des études en pharmacogénomique, ainsi que celui de connaître de génotype des patients, afin d'éviter ces événements indésirables.

Au cours de ces dernières années, l'amélioration des techniques de biologie moléculaire, la diminution drastique de leurs coûts, ainsi que des études toujours plus nombreuses sur les relations entre gènes et médicaments ont fait émerger la notion de médecine personnalisée. Il ne s'agit plus maintenant de développer et commercialiser un médicament pour une pathologie donnée, mais également pour une population bien définie.

La mise en oeuvre des connaissances acquises lors des études pharmacogénomiques dans la pratique clinique est de plus en plus fréquente, notamment dans le cas des traitements de certains cancers, dont l'efficacité est fortement liée aux génotypes des patients. La Figure 2 présente une liste non exhaustive de médicaments mis sur le marché aux États-Unis, prenant en compte des facteurs génétiques dans leur indication même.

Agent	Variant Genes	Proposed Outcome
Tamoxifen	CYP2D6 Poor Metabolizers	Reduced survival Decreased time to progression Decreased hot flashes
	CYP2D6 Extensive Metabolizers	Increased likelihood of tamoxifen discontinuation
	<i>UGT2B15</i>	Increased risk of recurrence
	<i>SULT1A1</i>	Increased risk of recurrence Inferior survival
	<i>ABCB1</i>	Shorter time to recurrence
	<i>ESR1</i>	Increased risk of relapse
	<i>ESR2</i>	Decreased hot flashes
Aromatase Inhibitors	<i>CYP19</i>	Improved time to progression Poor response to letrozole
	<i>TCL1A</i>	Increased musculoskeletal adverse events
	<i>ESR1</i>	Increased risk of discontinuation due to toxicity
	<i>CYP19A1</i>	Increased risk of discontinuation due to toxicity
	<i>NCOR1</i>	Decreased risk of discontinuation of letrozole
Trastuzumab	Fc gamma RIIIa-158 V/V	Improved objective response rate & progression-free survival

FIGURE 2 – Médicaments approuvés par la FDA dont la notice contient des éléments concernant la génétique au sein de leur indication [19].

La découverte de nouvelles connaissances en pharmacogénomique est complexe. D'une part, la rareté des variants génétiques impliqués rend difficile le recrutement d'un nombre suffisant de patients porteurs des variants considérés pour la réalisation d'études cliniques. D'autre part, la réponse aux médicaments peut dépendre de plusieurs variants, mais également de facteurs non génétiques tels que l'alimentation et l'environnement. Cet aspect multifactoriel complexifie la recherche de relations fortes entre variants génétiques et réponses spécifiques à un traitement. Afin de mieux comprendre comment les gènes peuvent modifier l'action d'un médicament, il convient de développer les mécanismes mis en oeuvre entre l'absorption et

l'action d'un médicament.

## **1.1 Le métabolisme et l'action du médicament**

### **1.1.1 La pharmacocinétique**

La pharmacocinétique est l'étude du devenir des médicaments suite aux actions de l'organisme, de l'administration à la cible pharmacologique. Les processus pharmacocinétiques sont des processus complexes, qui sont classiquement subdivisés en 4 sous-processus : l'absorption, la distribution, le métabolisme et l'excrétion du médicament.

#### **Absorption**

Au cours de la phase d'absorption, les substances actives passent du site d'absorption (voie orale, intraveineuse, rectale, etc.) à la circulation générale. Au cours de cette phase, une certaine quantité de substance active peut être perdue. La fraction de médicament arrivant bien sur son site d'action, ainsi que la vitesse à laquelle elle y parvient correspond à la biodisponibilité du médicament. Celle-ci dépend de nombreux paramètres, notamment de la voie d'administration. En effet, certaines voies, telles que la voie orale, nécessitent le passage par de nombreux compartiments, de nombreuses membranes biologiques, et sont liées à un premier passage hépatique au cours duquel de nombreuses enzymes présentes au niveau du foie dégradent les substances administrées. L'ensemble de ces contraintes diminue drastiquement la biodisponibilité de certains médicaments. Certains types de substances, telles que les protéines, ne peuvent d'ailleurs pas, pour ces raisons, être administrées par voie orale [20].

#### **Distribution**

Une fois arrivées dans le compartiment sanguin, la (ou les) molécule(s) active(s) sont transportées vers leur cible. Ce transport se fait via la fixation à des protéines plasmatiques (albumine, lipoprotéines, etc.), avec une force de fixation et une réversibilité variables, en fonction de la substance. L'intégralité de la quantité passée dans la circulation ne se fixe cependant pas. On distingue une fraction liée, fixée aux protéines plasmatiques et piégée

dans la circulation, et une fraction libre, qui représente la fraction active du médicament dans le compartiment.

## Métabolisme

Le métabolisme d'un médicament est l'ensemble des étapes de biotransformation au cours desquelles ce dernier est transformé par le système enzymatique de l'organisme en un ou plusieurs composés, les métabolites. Ces produits de métabolisme peuvent avoir une activité pharmacologique [21], mais peuvent aussi être inactifs [22], voire toxiques [23]. L'organe ayant l'activité métabolique principale est le foie, où se concentre un grand nombre d'enzymes impliquées dans le métabolisme des xénobiotiques. Deux grands types de réactions sont impliquées dans le métabolisme des substances actives : les réactions de Phase I et de Phase II, décrites dans les paragraphes suivants.

### Les réactions de phase I

On distingue trois types de réactions de phase I : les réactions d'oxydation, les réactions de réduction, et les réactions d'hydrolyse. Souvent catalysées par des enzymes, elles conduisent à la production de dérivés hydroxylés, carboxylés ou aminés.

Les cytochromes P450 – nommées ainsi de par leur pic d'absorption en spectrophotométrie [24] – sont un groupe d'enzymes, plus spécifiquement de mono-oxygénases, d'une importance majeure dans les processus de phase I, et plus généralement dans le métabolisme des xénobiotiques. En 2017, 19 familles de cytochromes P450 sont identifiées [25]. Toutefois, certaines familles de cytochromes sont plus particulièrement impliquées dans le métabolisme des médicaments chez l'humain (le CYP3A4/5/7, le CYP2D6, etc.).

Au vu de leur impact majeur dans le métabolisme des médicaments, les facteurs influant sur l'action des cytochromes peuvent modifier de manière conséquente les effets thérapeutiques d'une molécule. Un exemple typique est celui de l'inhibition du CYP3A4 par le jus de pamplemousse [26].

Les facteurs génétiques jouent un rôle prépondérant dans l'action des enzymes de la famille des cytochromes P450, et donc dans la réponse aux médicaments. De très nombreux variants des cytochromes P450 existent, entraînant une grande variabilité individuelle ou populationnelle dans l'action des substances thérapeutiques. Par exemple, les populations asiatiques métabolisent peu, relativement aux autres populations, les substances dont les voies passent par le CYP2C19 [27], [28], telles que les inhibiteurs de pompes à proton, ou encore la phénytoïne. Ce métabolisme particulier est lié aux variants génétiques fréquents dans ces populations.

La découverte de ces variabilités génétiques a entraîné un changement de paradigme : d'une médecine globale, où un médicament est censé traiter une maladie quel que soit le patient, une nouvelle médecine est née : la médecine personnalisée, où un traitement est adapté à un individu donné.

## **Les réactions de phase II**

Les produits de la phase I subissent par la suite des réactions de conjugaison donnant lieu à des composés hydrosolubles, et ainsi aisément éliminables dans les processus d'excrétions. Ces réactions existent afin de protéger l'organisme contre les xénobiotiques, mais peuvent jouer un rôle néfaste dans le cas où lesdits xénobiotiques dont l'organisme essaie de se protéger sont des substances à but thérapeutique.

## **Excrétion**

L'excrétion, ou élimination, consiste en l'évacuation des xénobiotiques hors de l'organisme, le plus souvent par élimination rénale, via l'urine, et hépatique, via la bile. Certaines substances peuvent être cependant excrétées par la salive, le lait maternel, ou encore la sueur, où l'on peut les retrouver.

### **1.1.2 La pharmacodynamique**

La pharmacodynamique est l'étude de l'action du médicament sur le corps, plus précisément des mécanismes impliqués dans son action thérapeutique. Dans le cas le plus courant,

le mécanisme d'action d'une substance thérapeutique passe par sa fixation spécifique à une protéine, pouvant être un transporteur, une enzyme, ou, le plus souvent, un récepteur.

Un récepteur est une macromolécule localisée au niveau de la membrane plasmique, du cytoplasme ou du noyau d'une cellule, permettant de moduler des actions *via* la fixation d'une molécule spécifique au domaine de ce récepteur. Cette molécule spécifique à un récepteur donné est appelée *ligand*. Le site actif d'un récepteur est classiquement vu comme une serrure, le médicament étant la clé. Cette analogie montre bien la notion de site spécifique, de complémentarité ligand-récepteur, et d'action subséquente à la fixation du ligand sur son récepteur.

Quatre grandes familles de récepteurs sont particulièrement étudiées :

- Les récepteurs couplés aux protéines G (RCPG),
- Les récepteurs Tyrosine Kinase / Guanylate Cyclase,
- Les récepteurs canaux ioniques,
- Les récepteurs nucléaires.

L'affinité qu'ont les médicaments pour ces macromolécules étant intimement liée à leur conformation spatiale, et donc *in fine* à leur structure, les gènes impliqués dans leur fabrication sont donc des points critiques quant à l'action des substances thérapeutiques. Les variations de ces gènes peuvent avoir une influence majeure sur un effet thérapeutique spécifique.

Dans certains cas, des mutations peuvent entraîner une modification de la conformation du récepteur favorable à un traitement, comme c'est le cas par exemple pour certaines mutations de l'*EGFR*, liées à une meilleure efficacité du gefitinib dans le traitement du cancer du poumon à petites cellules [29]. Cependant, une variation peut avoir des effets délétères pour le patient, par exemple en diminuant l'effet d'un traitement [30]. Ces mutations sont fréquemment des *SNP* (*Single-Nucleotide Polymorphism*), ne concernant qu'une seule paire de bases.

### 1.1.3 Défis de la pharmacogénomique

Nous avons vu dans les deux points précédents l'importance de la génétique dans la réponse aux médicaments. Les changements dans le métabolisme ou la cible d'une substance peuvent entraîner des changements drastiques de ses effets qui ne peuvent être éludés. L'étude de l'impact des variations de gènes bien particuliers sur l'effet thérapeutique attendu est une nécessité.

C'est pour cette raison que la pharmacogénétique prend une place de plus en plus importante au sein des études – tant cliniques que fondamentales – et dans la mise en place même des traitements et protocoles de soin. Un nombre croissant de protocoles de traitement de pathologies cancéreuses prend en compte les facteurs génétiques afin de mettre en place le protocole de soin le plus adapté à un patient donné [31].

L'intérêt d'accélérer les processus de découverte des variations ayant un impact sur l'action des médicaments est donc d'une importance majeure, à l'heure où de plus en plus de pathologies sont concernées par la personnalisation des traitements.

## 2 L'extraction de connaissances à partir des bases de données

### 2.1 Définition

Le processus de fouille de données – ou *data mining* –, vise à extraire des connaissances à partir de données. Il convient dans un premier temps de faire la distinction entre données, informations et connaissances. Bien que parfois utilisés de manière interchangeable, ces trois termes ont un sens bien différent [32]. Les données sont un ensemble de signes, associés à une syntaxe propre qui décrit une observation du monde. Elles permettent d'enregistrer cette observation et ses attributs intrinsèques. Ces données deviennent informations lorsqu'elles sont interprétées, c'est à dire lorsqu'un sens leur est associé. Enfin, il est question de connaissances lorsque les informations sont représentées et assimilées de sorte à pouvoir être réutilisées.



Un ensemble de méthodes et d’algorithmes est à la disposition des *data scientists* – personnes en charge de conduire les processus d’ECBD – afin de mener à bien cette tâche, en fonction des données et de l’objectif à atteindre. Une description de certaines de ces méthodes sera faite au sein de ce chapitre.

## 2.2 Le processus de fouille de données

Plusieurs étapes sont nécessaires afin de transformer les données en connaissances. Ces étapes ont été formalisées par Fayyad *et al.* en 1996, et sont présentées Figure 3.

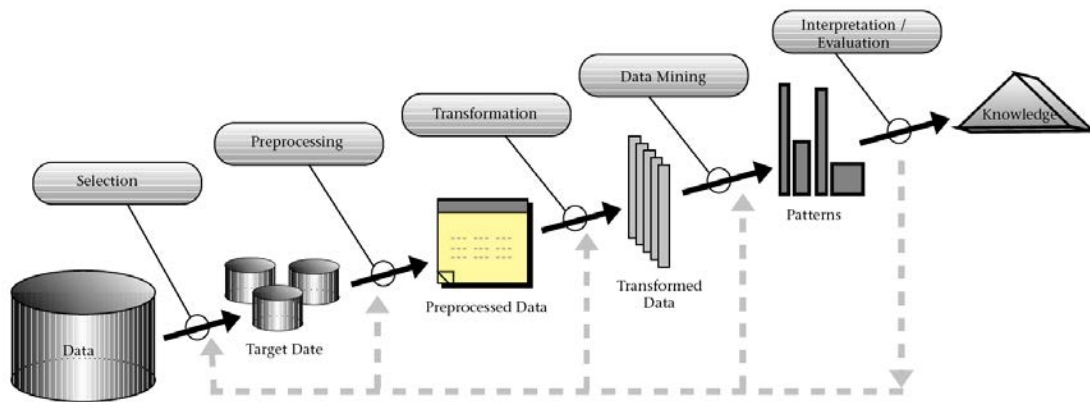


FIGURE 3 – Processus de KDD (Knowledge Discovery in Databases) [33].

La première étape consiste en la sélection des données. Il s’agit de déterminer les sources de données d’intérêt pour l’étude considérée. Cette étape est déterminante, dans la mesure où les résultats dépendent nécessairement des données initiales.

L’étape suivante, le preprocessing, consiste en la préparation des données, en vue de les traiter. Les différents jeux de données initiaux sont ici regroupés, une stratégie de gestion des données manquantes le cas échéant est mise en place, ainsi qu’une élimination des données bruitées si cela est possible.

Une fois les données prétraitées, vient l’étape de transformation. Il s’agit principalement d’adapter les données à la méthode de fouille subséquente. Nous verrons dans les chapitres

suivants que ces dernières sont nombreuses, et sont différentes sur de nombreux points, notamment concernant les données qu'elles acceptent en entrée. Il est donc nécessaire d'adapter les données afin qu'elles puissent être fouillées avec la méthode choisie. On comprend ici l'intérêt d'avoir dès le départ une stratégie de fouille claire. D'autre part, il est également possible de transformer les données via la sélection d'attributs, ou réduction de dimensionnalité, afin d'améliorer les performances du modèle.

Les données prétraitées et transformées peuvent enfin être utilisées par la méthode de fouille choisie, afin d'en extraire des régularités, et *in fine*, des connaissances.

On distingue parmi les nombreux algorithmes de Fouille de données existants deux catégories de méthodes : les méthodes non supervisées et les méthodes supervisées. Nous allons ici décrire les différences entre ces deux catégories, en présentant des exemples d'algorithmes y appartenant. Cette description a pour but de fournir une introduction aux concepts clés, sans être exhaustive.

## 2.3 Les méthodes non supervisées

Les algorithmes appartenant à cette catégorie ont pour objectif de mettre en évidence des structures ou motifs homogènes au sein de données hétérogènes, sans connaissance à priori de la classe d'appartenance (ou étiquette) de chaque objet. Un exemple classique de méthode non supervisée est le **clustering**. De façon générale, les algorithmes de *clustering* ont pour objectif de grouper les objets au sein de **clusters**, de manière à ce que la similarité entre individus d'un même cluster soit plus forte que la similarité entre objets de clusters distincts. On parle de similarité *intra-cluster* et *inter-cluster*, respectivement. Nous allons ici décrire deux méthodes de clustering courantes dans la littérature : la méthode des k-moyennes, et le clustering hiérarchique.

### 2.3.1 La méthode des k-moyennes

La méthode des k-moyennes, développée par Mac Queen *et al.* [34] en 1967, est l'une des méthodes les plus courantes d'apprentissage non supervisé. Il s'agit d'une méthode basée sur

les barycentres, où les clusters sont définis par leurs vecteurs centraux  $\mu_i$  – leurs barycentres –, ne faisant pas nécessairement partie des données. L’objectif est de regrouper  $n$  observations  $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$  en  $k$  clusters  $C = \{C_1, \dots, C_k\}$ , en cherchant à minimiser la distance entre les observations associées à chaque cluster.

Ici,  $\mathbf{x}_1, \dots, \mathbf{x}_n$  représentent des vecteurs de dimension  $d$ , et  $\mu_i$  la moyenne des points dans le cluster  $C_i$ . Il s’agit donc d’une tâche d’optimisation de fonction, avec la fonction à minimiser étant la somme des carrés des distances intra-clusters (ou *WCSS*, *Within Cluster Sum of Squares*).

$$WCSS = \sum_{i=1}^k \sum_{\mathbf{x}_j \in C_i} \|\mathbf{x}_j - \mu_i\|^2 \quad (1)$$

L’algorithme des K-moyennes est présenté dans l’algorithme 1.

---

**Algorithm 1** Algorithme des k-moyennes

---

```

1 : procedure KMEANS(D, K)
2 :   Initialisation de  $k$  vecteurs  $\mu_1, \dots, \mu_k \in \mathbb{R}^d$ 
3 :   Tant que Non convergence des  $\mu_1, \dots, \mu_k$  Faire
4 :     Assigner chaque vecteur  $\mathbf{x} \in D$  à  $\argmin_j d(\mathbf{x}, \mu_j)$ 
5 :     Pour  $j = 1$  à  $k$  Faire
6 :        $Cluster_j \leftarrow \{\mathbf{x} \in D \mid \mathbf{x} \text{ assigné au cluster } j\}$ 
7 :        $\mu_j = \frac{1}{|D_j|} \sum_{\mathbf{x} \in D_j} \mathbf{x}$ 
8 :     Fin Pour
9 :   Fin Tant que
10 :   Retourner  $\mu_1, \dots, \mu_k$ 
11 : Fin procedure
```

---

**Initialisation** La première étape est l’étape d’initialisation, où  $k$  points  $\mu_1, \dots, \mu_k \in \mathbb{R}^d$  sont initialisés et sont considérés comme les centres des  $k$  clusters. Plusieurs méthodes d’initialisation existent. L’approche la plus naïve consiste à effectuer l’initialisation des  $k$  points dans l’espace de dimension  $d$  de manière aléatoire. D’autres méthodes d’initialisation peuvent être utilisées. Thiesson *et al.* [35] présentent une méthode consistant à prendre la moyenne de l’ensemble des points, puis perturber cette valeur  $K$  fois de manière aléatoire. L’approche Forgy [36], consiste à choisir  $K$  points du jeu de données aléatoirement en tant que barycentres initiaux. L’approche de Kaufman, présentée par Kaufman et Rousseeuw [37], est une

approche permettant une initialisation des barycentres de manière déterministe. Une étude empirique des différentes méthodes d'initialisation est présentée dans [38]. Les auteurs ont montré de meilleures performances vis-à-vis de la robustesse avec l'initialisation de Kaufman. Cependant, l'initialisation aléatoire, souvent utilisée en pratique, montre de bonnes performances. Il est toutefois nécessaire de réitérer l'opération plusieurs fois pour être sûr de ne pas être dans un minimum local.

**Assignment** Chacun des points  $\mathbf{x} \in D$  est par la suite associé au cluster  $j$  dont le barycentre  $\mu_j$  est le plus proche, *i.e.*, dont la distance euclidienne est la plus faible. Enfin, une fois chacune des observations associées au cluster le plus proche, les barycentres sont recalculés. Il s'agit des lignes 5 à 7 de l'algorithme. Cette procédure d'assignation de points au cluster dont le barycentre est le plus proche et de recalcul de ces derniers est répétée jusqu'à convergence. On considère que l'algorithme converge lorsque la fonction de coût (en l'occurrence, la somme des carrés des distances intra-clusters) n'est plus modifiée d'une itération à une autre, ou que cette modification est inférieure à un certain seuil. Cette convergence est assurée, mais, dans certaines conditions, l'algorithme peut converger vers un minimum local [39]. On peut aussi considérer comme critère d'arrêt le fait qu'aucun point ne change de cluster entre deux itérations.

La Figure 4 présente un exemple de déroulement de l'algorithme.

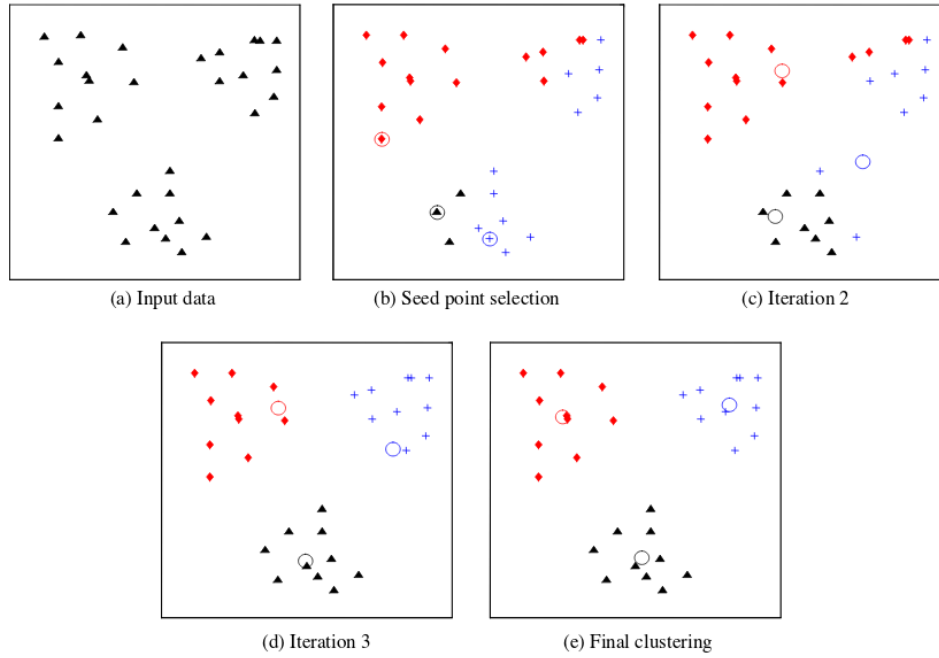


FIGURE 4 – Ensemble d'étapes de la méthode des k-moyennes [40] (a) Données bidimensionnelles. (b) Trois barycentres initiaux (entourés) sont sélectionnés, et une première assignation est réalisée. (c) et (d) sont les itérations subséquentes, où les barycentres et l'assignation des points à chaque cluster sont réévalués. (e) Clusters à convergence.

Bien que très largement utilisé en pratique, l'algorithme des K-moyennes présente quelques limites. D'une part, l'un de ses paramètres est le nombre de clusters recherchés. Une connaissance, ou, à minima, une intuition concernant la répartition des individus est nécessaire. D'autre part, une supposition forte est faite concernant la forme des clusters. En effet, la méthode des k-moyennes, à convergence, ne retourne que des partitions convexes de l'espace, de part l'utilisation de la distance euclidienne.

**Définition** Dans un espace euclidien, une région convexe est une région où tout segment entre deux points de la région est aussi à l'intérieur de la région. Un exemple de région convexe est présenté Figure 5.

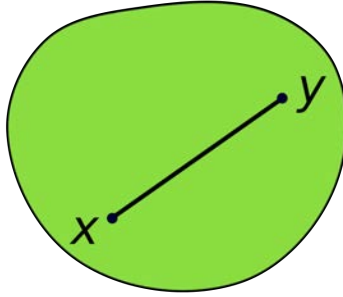


FIGURE 5 – Exemple de région convexe dans un espace à deux dimensions. [41]

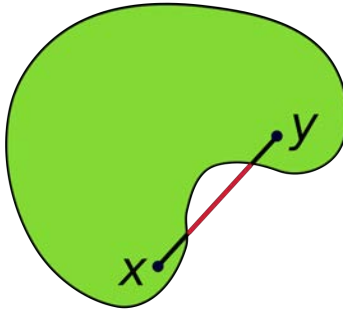


FIGURE 6 – Exemple de région non convexe [41]. Il est possible de tracer des segments entre deux points de la région qui ne sont pas intégralement à l'intérieur de la région.

Ainsi, dans le cas où les données ne constituent pas des ensembles convexes, la méthode ne donnera pas de résultats satisfaisants.

### 2.3.2 Le clustering hiérarchique

Un deuxième type de méthodes non supervisées sont celles de clustering hiérarchique. Il s'agit de méthodes visant à obtenir une hiérarchie de clusters, constituant un arbre binaire, le **dendrogramme**, dont un exemple est présenté Figure 7.



FIGURE 7 – Dendrogramme représentant les différences génétiques entre 17 populations sur la base des différences de séquences entre 15 loci [42]

Pour ce faire, deux approches sont possibles :

- L'approche **agglomérative**, où, à l'état initial, où chaque objet est associé à cluster propre, et où sont fusionnées des paires de clusters à chaque itération
- L'approche **divisive**, où tous les objets font partie du même cluster à l'état initial, avec une division des clusters à chaque itération. Dans les deux cas, chaque itération correspond à un nouveau niveau du dendrogramme, niveau qui correspond à un nombre de clusters spécifique.

Afin de déterminer quels objets séparer ou quels clusters fusionner, deux métriques sont nécessaires : une mesure de la distance entre deux objets, et une autre calculant la distance entre ensembles d'objets.

**Choix de la métrique** De très nombreuses mesures de distances existent. Le domaine est si prolifique que certains ouvrages dédiés existent, tels que l'encyclopédie des distances,

consacrant près de 700 pages au sujet [43]. On retrouve par exemple :

$$\text{La distance euclidienne : } d(x, y) = \|x - y\|_2 = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2)$$

$$\text{La distance de Manhattan : } d(x, y) = \|x - y\|_1 = \sum_{i=1}^n (|x_i - y_i|) \quad (3)$$

$$\text{La distance euclidienne au carré : } d(x, y) = \|x - y\|_2^2 = \sum_{i=1}^n (x_i - y_i)^2 \quad (4)$$

Ces métriques sont adaptées à des données numériques continues. Dans le cas de données non numériques, d'autres mesures, telles que celles de Levenshtein [44] ou de Hamming [45], existent.

**Choix du critère de regroupement** Ce critère de regroupement (ou *linkage criterion*) permet de déterminer la distance entre ensembles d'objets. On retrouve :

$$\text{Complete linkage : } \max\{d(x, y) : x \in C_1, y \in C_2\} \quad (5)$$

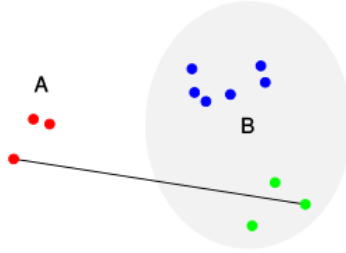


FIGURE 8 – Représentation schématique de la distance entre deux clusters A et B, selon le critère de *complete linkage*. [46]

$$\text{Average linkage : } \frac{1}{|C_1||C_2|} \sum_{x \in C_1} \sum_{y \in C_2} d(x, y) \quad (6)$$



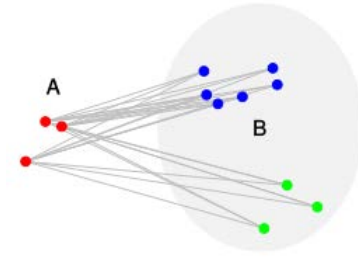


FIGURE 9 – Représentation schématique des distances entre deux clusters A et B, avec le critère de l'*average linkage* [47]

$$\text{Single linkage} : \min\{d(x, y) : x \in C_1, y \in C_2\} \quad (7)$$

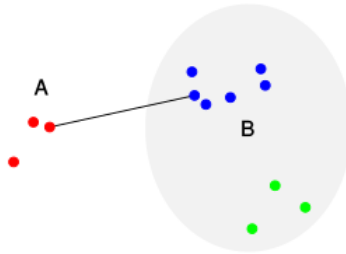


FIGURE 10 – Représentation schématique de la distance entre deux clusters A et B, selon le critère du *single linkage* [48]

Les avantages de ces méthodes sont :

- le nombre de clusters n'est pas requis afin d'obtenir la hiérarchie de clusters
- la hiérarchie obtenue peut-être facilement visualisée, et peut fournir des informations intéressantes sur les données

Cependant, le choix de la mesure de distance entre objets et du critère de regroupement peuvent avoir une grande influence sur les résultats obtenus. Les méthodes de clustering hiérarchique sont couramment utilisées, notamment dans les études d'expression, où elles sont très populaires, afin de regrouper les gènes dont les profils d'expression sont similaires.

## 2.4 Les méthodes supervisées

Nous avons vu précédemment les méthodes d'apprentissage non supervisé, où la classe d'appartenance des objets n'est pas connue à priori (on parle alors de **données non étiquetées**). Les méthodes d'Apprentissage supervisé quant à elles visent à apprendre une fonction à partir de données étiquetées, dans l'objectif d'inférer la classe d'objets nouveaux. L'apprentissage du modèle se fait au cours d'une première étape, sur des données d'entraînement, le *training set*. Le modèle est ensuite évalué sur un jeu de données indépendant des données d'apprentissage, le *test set*.

Formellement, soit un ensemble d'objets  $\{(x_1, y_1), \dots, (x_n, y_n)\}$ , où  $x_i$  représente le vecteur d'attributs du  $i^{me}$  individu, et  $y_i$  sa classe d'appartenance. Un algorithme d'Apprentissage supervisé cherche une fonction  $f : X \rightarrow Y$ , où  $X$  représente l'espace d'entrée, et  $Y$  l'espace de sortie. L'espace d'entrée est un espace de même dimension que les vecteurs d'attributs. Nous allons décrire dans ce chapitre trois méthodes d'Apprentissage supervisé : les arbres de décisions, les forêts d'arbre aléatoires, et enfin, les machines à vecteur de support.

### 2.4.1 Les arbres de décision

Les méthodes basées sur les arbres de décisions sont très populaires, de par notamment leur expressivité. En effet, les modèles obtenus peuvent être représentés graphiquement, et permettre leur interprétation. Nous traiterons ici des arbres de décisions utilisés en vue d'effectuer une classification. Afin de comprendre leur fonctionnement, commençons par un exemple.

Supposons que le problème posé soit celui de la prédiction du fait qu'un match soit joué ou non. Les attributs connus sont :

- La température, prenant une valeur parmi {Faible, Elevée},
- La présence de vent, prenant une valeur parmi {Oui, Non}.

La classe à prédire est ici un booléen : soit le match a lieu, soit il n'a pas lieu, prenant ainsi une valeur parmi {Oui, Non}.

Soit le jeu de données présenté Table 1.

TABLE 1 – Jeu de données d’apprentissage

<b>Temperature</b>	<b>Vent</b>	<b>Classe</b>
Elevée	Non	Oui
Faible	Non	Oui
Elevée	Non	Oui
Faible	Oui	Non
Faible	Non	Oui
Elevée	Oui	Non
Elevée	Non	Oui

Un arbre de décision a une structure d’arbre, où les noeuds représentent un test sur un attribut, une branche une division sur un attribut en fonction de sa valeur, et une feuille une classe. Afin de construire cet arbre, l’approche choisie est très souvent une approche descendante, où l’on choisit à chaque étape l’attribut séparant le mieux l’ensemble des objets. Afin de déterminer cet attribut, plusieurs métriques existent. L’une des plus courantes est la mesure du **gain d’information**, basée sur le concept **d’entropie**, adapté par Shannon à la théorie de l’information.

Soit  $H(X)$  l’entropie définie par l’équation :

$$H(X) = - \sum_{i=1}^n p_i \log(p_i) \quad (8)$$

où  $X$  est la variable aléatoire représentant la classe à prédire et pouvant prendre  $n$  valeurs, et les  $p_i$  représentent les pourcentages de chaque classe présente dans le jeu de données. Cette mesure quantifie l’homogénéité d’un ensemble. L’entropie est maximale lorsqu’aucune information sur le système n’est disponible.

Le gain d’information (noté  $GI$  par la suite) est obtenu par la relation :

$$GI(X, a) = H(X) - H(X|a) \quad (9)$$

, où  $a$  représente un attribut.

Un bon attribut est un attribut qui contient beaucoup d'information, *i.e.*, qui diminue l'entropie. Dans notre exemple, nous avons  $H(X) = -\frac{5}{7}\log(\frac{5}{7}) + \frac{2}{7}\log(\frac{2}{7}) = 0.863$ . Calculons la quantité d'information de l'attribut *Température*. Pour obtenir la valeur de  $H(X|Température)$ , il est nécessaire de diviser le jeu de données en fonction de cet attribut, comme cela est fait Figure 11.

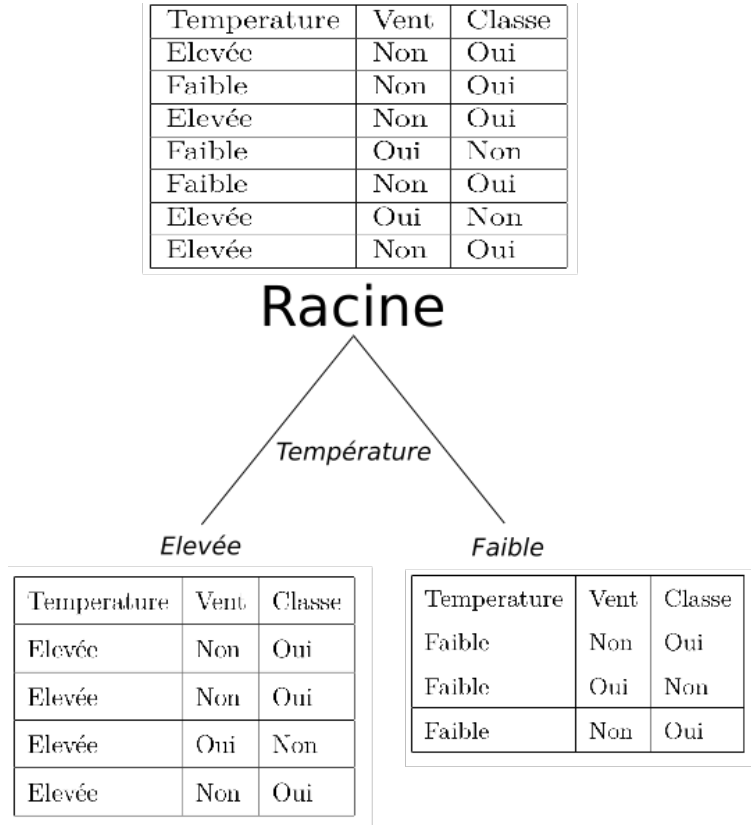


FIGURE 11 – Division du jeu de données en fonction de l'attribut *Température*.

Chaque branche a maintenant son entropie propre.

$$H(fils\_gauche) = -\frac{3}{4}\log(\frac{3}{4}) + \frac{1}{4}\log(\frac{1}{4}) = 0.811 \quad H(fils\_droit) = -\frac{1}{3}\log(\frac{1}{3}) + \frac{2}{3}\log(\frac{2}{3}) = 0.918 \quad (10)$$

$H(X|Température)$  correspond à la somme pondérée des entropies des noeuds fils :

$$H(X|Température) = -\frac{4}{7}H(fils\_gauche) + \frac{3}{7}H(fils\_droit) \quad (11)$$

Le gain d'information, correspondant à la contribution de l'attribut à la réduction d'entropie,

est ainsi obtenu par la différence  $H(X) - H(X|Temperature) = 0,006$ . Ici, ce gain d'information est très faible. Cela se comprend au vu des données. En effet, on observe facilement que l'attribut température n'a, *à priori*, pas beaucoup d'influence sur la classe à prédire.

Afin de construire un arbre de décision, il est nécessaire de comparer le gain d'information associé de chaque attribut, et effectuer la division du jeu de données sur l'attribut associé au gain d'information le plus élevé. Le processus est répété jusqu'à obtention d'un arbre dont les feuilles sont pures (présence d'une seule classe). Le critère d'arrêt peut être différent, en fonction des usages. Le processus peut par exemple s'arrêter à un certain niveau de l'arbre, ou encore lorsque le gain d'information est nul. Dans notre exemple, le premier noeud serait donc une division sur l'attribut *Vent*, comportant plus d'information que l'attribut *température*.

Un exemple d'arbre de décision est présenté Figure 12.

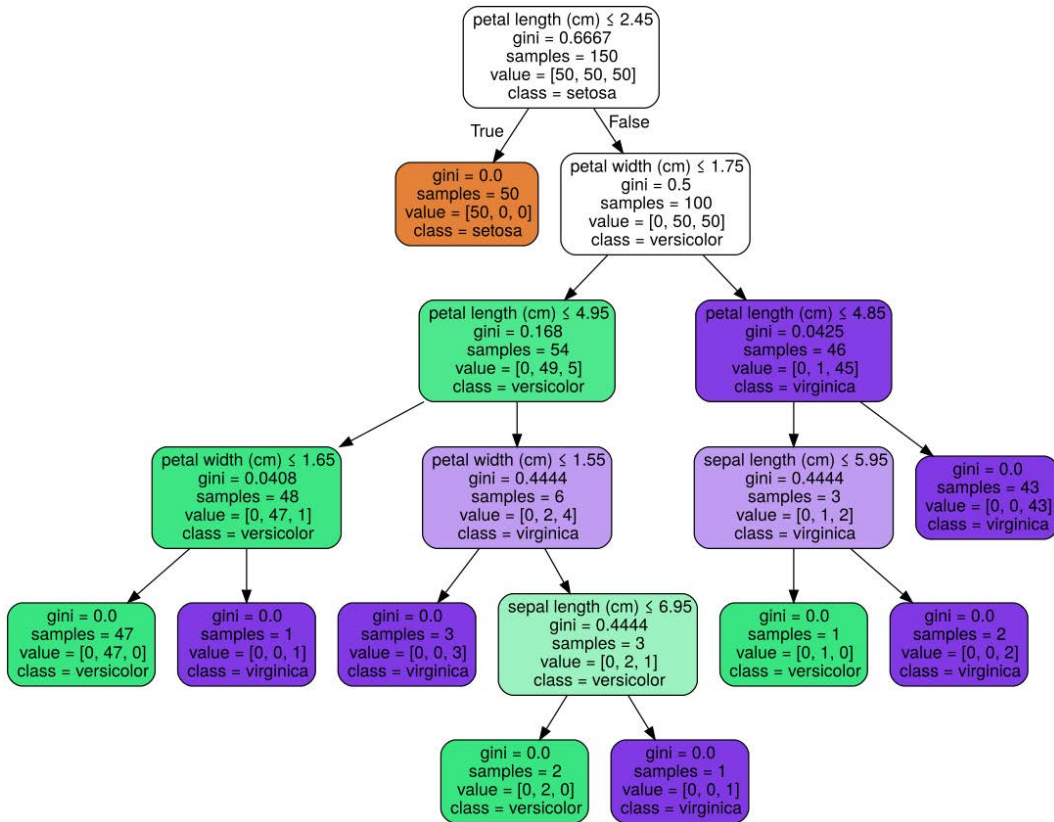


FIGURE 12 – Exemple d'arbre de décision, dans le cadre de la détermination d'espèces de fleurs en fonction de leurs attributs morphologiques [49].

Parmi les avantages des arbres de décisions, on retrouve :

- la capacité de prendre en compte à la fois des données numériques et catégorielles,
- la visualisation, et l'interprétation aisée des résultats,
- la similarité avec les processus décisionnels *humains*.

Cependant, les performances et la robustesse des arbres de décision seuls sont de manière générale inférieures à d'autres méthodes, telles que les forêts d'arbres aléatoires [50].

#### 2.4.2 Les forêts d'arbres aléatoires

La forêt d'arbres aléatoire (ou *Random Forest*) est une famille de méthodes d'ensemble, dont le principe général est de construire une collection de prédicteurs, pour ensuite agréger l'ensemble de leurs prédictions. Cet algorithme, créé par Leo Breimann, consiste en la création d'une multitude d'arbres de décision en fonction d'un ensemble de données d'entraînement qui voteront pour classer de nouvelles données [51]. Cet algorithme est l'un des plus populaires dans la littérature et dans les applications de l'apprentissage automatique. Le périphérique Kinect de Microsoft, notamment, utilise un type de forêt aléatoire pour son algorithme de reconnaissance de mouvement [52]. L'algorithme général se décompose ainsi [53] [51] :

---

**Algorithm 2** Algorithme de construction d'une forêt aléatoire.

---

```
1 : procedure RANDOMFOREST(NBARBRES,NBFEATURES,TAILLEECHANTILLON
2 : ,ENSEMBLEENTRAINEMENT)
3 :   Pour chaque arbre Faire Construire un échantillon bootstrap en tirant
      TailleEchantillon fois, avec remise, dans l'ensemble d'entraînement.
4 :     Pour chaque nœud de l'arbre Faire
5 :       Choisir NbFeatures variables sur lesquelles baser la décision au nœud.
6 :       Calcul et mise en place de la meilleure coupure basée sur ces NbFeatures
      variables dans l'échantillon bootstrap.
7 :     Fin Pour
8 :   Fin Pour
9 : Fin procedure
```

---

La méthode consiste en la création de plusieurs ensembles d'apprentissages, provenant d'un échantillon de l'ensemble de données initial. Ces ensembles sont décrits par un échantillon de leurs descripteurs (leurs *attributs*), et par leur classe d'appartenance. Un arbre de décision est par la suite entraîné sur chacun de ces sous-ensembles. On obtient donc plusieurs arbres

de décisions, qui prendront part à un vote donnant le résultat du modèle.

Cette approche permet de meilleures performances et une plus grande généralisation du modèle, tout en contrebalançant l'instabilité des arbres de décision simple, une forêt d'arbre étant moins sensible au bruit.

### 2.4.3 Les machines à vecteur de support

La méthode des machines à vecteurs de supports vise à séparer un ensemble d'objets en deux partitions – leurs classes, en traçant un hyperplan entre ces deux sous-ensembles. Il est donc, au vu de cette définition, essentiel que ces ensembles d'objets soient linéairement séparables.

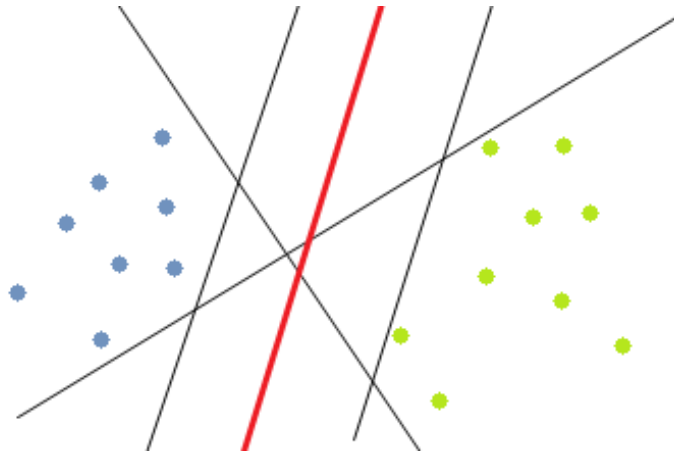


FIGURE 13 – Ensemble d'objets appartenant à une classe, *vert* ou *bleu*. Dans cet exemple, les objets sont représentés dans un espace bidimensionnel, où les hyperplans sont donc des droites. On remarquera que plusieurs hyperplans permettent de séparer les deux classes [54].

Bien que plusieurs hyperplans permettent de séparer les objets en deux classes, tous ne permettent pas d'obtenir les mêmes résultats. En effet, certains hyperplans, bien que séparant les objets d'apprentissage, ne donneront pas de bons résultats sur de nouveaux exemples : le modèle ne sera pas généralisable.

Intuitivement, on considère qu'un *bon* hyperplan séparateur est celui dont la distance avec les objets les plus proches dans chacune des classes –les **vecteurs supports**– est maximale. Cette distance est appelée **marge maximale**. Il s'agit donc d'un problème d'optimisation,

où l'on cherche à obtenir l'équation de l'hyperplan séparant les deux ensembles et pour lequel la marge est maximisée.

Un hyperplan correspond à l'ensemble des vecteurs  $\vec{x}$  satisfaisant l'équation :

$$\vec{w}\vec{x} + b = 0 \quad (12)$$

, où  $\vec{w}$  correspond à un vecteur normal à cet hyperplan. Afin de trouver celui pour lequel la marge est maximale, l'on construit deux hyperplans parallèles de chaque côté de ce dernier, d'équations :

$$\vec{w}\vec{x} + b = 1 \equiv \vec{w}\vec{x} + (b - 1) = 0 \quad (13)$$

$$\vec{w}\vec{x} + b = -1 \equiv \vec{w}\vec{x} + (b + 1) = 0 \quad (14)$$

On sait que la distance  $D$  entre deux hyperplans d'équations  $\vec{w}\vec{x} + b_1$  et  $\vec{w}\vec{x} + b_2$  est obtenue par la relation  $\frac{|b_1 - b_2|}{||\vec{w}||}$ , avec  $||\vec{w}||$  la norme de  $\vec{w}$ . Ainsi, les équations précédentes deviennent donc :

$$D = \frac{(b + 1) - (b - 1)}{||\vec{w}||} = \frac{2}{||\vec{w}||} \quad (15)$$

L'objectif est de maximiser  $D$ , et donc de minimiser  $||\vec{w}||$ . Il faut par ailleurs garder à l'esprit qu'une contrainte additionnelle est celle de la bonne classification. En effet, il faut que  $\vec{w}\vec{x} + b \leq -1$  si  $y_i = -1$ , et  $\vec{w}\vec{x} + b \geq 1$  si  $y_i = +1$ , avec  $y_i$  représentant la classe d'appartenance de l'objet  $i$ .



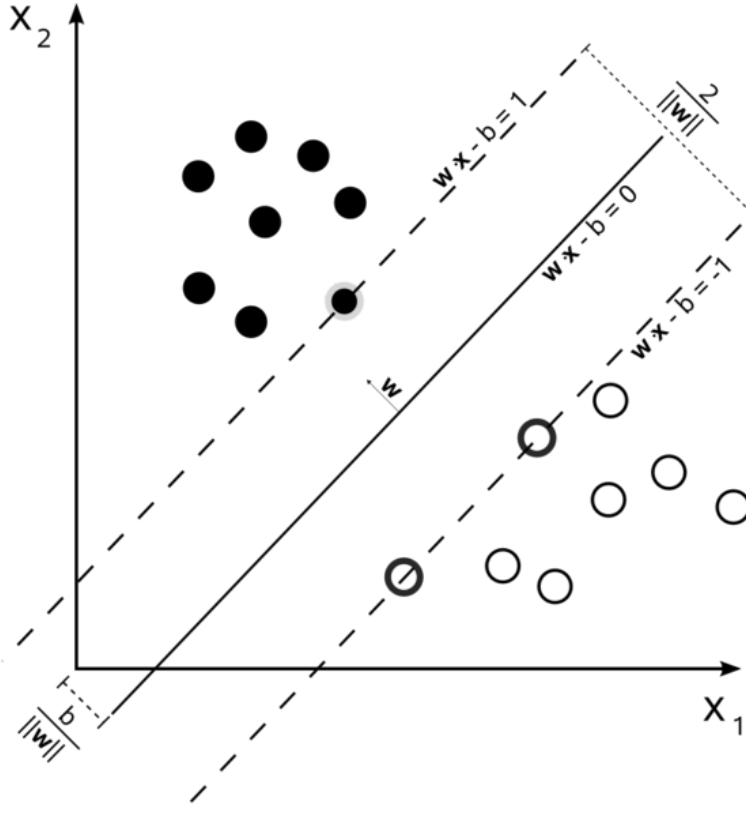


FIGURE 14 – Ensemble d’objets séparés par un hyperplan maximisant la marge, d’équation  $\vec{w}\vec{x} + b = 0$ . On remarquera que les frontières correspondent aux vecteurs supports [55].

Nous n’entrerons pas ici dans les détails concernant la résolution de ce problème d’optimisation. Cependant, certaines propriétés de ce problème et de ses solutions sont intéressantes. D’une part, l’optimisation de la marge – et donc, le calcul de l’hyperplan optimal – ne nécessite pas l’accès aux données complètes, mais uniquement aux produits scalaires entre les objets. Pour rappel, chaque objet est un vecteur de dimension  $n$ . D’autre part, le vecteur  $\vec{w}$  s’écrit comme une combinaison linéaire des vecteurs de supports. La marge ne dépend donc que de ces derniers.

Nous avons vu jusqu’à maintenant que les machines à vecteurs de supports permettent d’effectuer une classification que dans le cas où les données sont séparables de manière linéaire. La Figure 15 présente un exemple de ce type de cas.

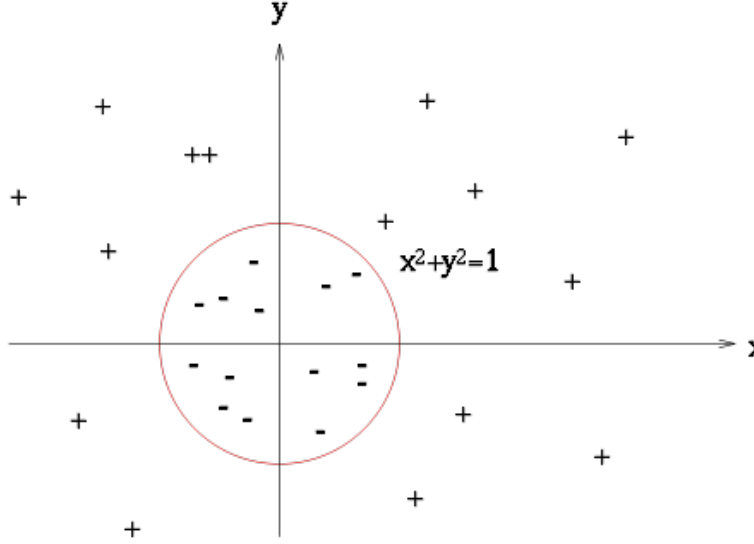


FIGURE 15 – Ensemble d’objets non séparables de manière linéaire. Ici, la frontière est circulaire [56].

Il est cependant possible d’utiliser la méthode des SVM dans ce cas, grâce à l’application de **l’astuce du noyau**. Avant de tenter de trouver un hyperplan séparateur, on passe d’abord dans un espace intermédiaire de dimension plus élevée, *via* une fonction  $\Phi : \mathbb{R}^d \rightarrow \mathbb{F}$ . Cependant, le calcul de cette transformation peut être computationnellement lourd.

Nous avons vu précédemment que l’expression du problème d’optimisation ainsi que sa solution ne dépendent que du produit scalaire  $\langle \vec{x}, \vec{x'} \rangle$  entre les objets. Il est possible d’éviter de calculer la représentation de chaque objet dans l’espace intermédiaire  $\mathcal{F}$ . Il s’agit de l’astuce du noyau suscitée. Une fonction **noyau**  $k : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$  est une fonction pouvant être exprimée en tant que produit scalaire dans un espace intermédiaire  $\mathbb{F}$ .

$$k(\vec{x}, \vec{x'}) = \langle \Phi(\vec{x}), \Phi(\vec{x'}) \rangle \quad (16)$$

Afin de mieux comprendre, prenons un exemple concret. Soient les deux objets  $x = \{x_1, x_2\}$  et  $y = \{y_1, y_2\}$ , et une fonction noyau  $k$  telle que :

$$k = (x^T y)^2 = (x_1 y_1 + x_2 y_2)^2 = (x_1^2, \sqrt{2} x_1 x_2, x_2^2)^T, (y_1^2, \sqrt{2} y_1 y_2, y_2^2) = \Phi(x)^T \Phi(y) \quad (17)$$

Cette fonction  $k$  définit de manière implicite le mapping  $\Phi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$  dans un espace de plus haute dimension,  $\mathbb{R}^3$ . De plus, elle permet le calcul du produit scalaire  $\langle \Phi(x), \Phi(y) \rangle$ . Toutes les fonctions noyaux ont ces propriétés. L'utilisation des noyaux permet de s'affranchir de la limitation des SVM, les données d'entrées ne devant plus être forcément linéairement séparables.

## 2.5 Les méthodes de fouille appliquées aux graphes

### 2.5.1 Les graphes et leur représentation

Un graphe simple  $G = (V, E)$  est défini par un ensemble de nœuds  $V = \{v_1, v_2, \dots, v_n\}$ , reliés deux à deux par des arêtes, appartenant à l'ensemble  $E$ . Ces arêtes peuvent être orientées — on parle alors de graphe *orienté* — ou non. Si le graphe est orienté, les arêtes sont aussi appelées *arcs*. Une arête entre un nœud  $v_i$  et un nœud  $v_j$  est notée  $e_{ij}$ . Un graphe peut être *pondéré*. Dans un tel graphe, une fonction de pondération  $\omega : E \rightarrow \mathbb{R}$  associe un poids à chaque arête. Un exemple de graphe simple non orienté est présenté Figure 16.

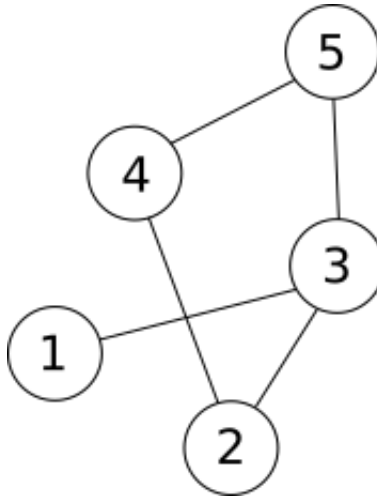


FIGURE 16 – Exemple de graphe simple.

On retrouve aussi la notion de graphe étiqueté. Dans ce type de graphe, les nœuds et les arêtes sont associés à un élément d'un ensemble fini de symboles (lettres, chaînes de caractères, etc.), *via* une fonction d'étiquetage. La densité d'un graphe simple non orienté est

donnée par la relation

$$Densit = \frac{2|E|}{|V|(|V| - 1)} \quad (18)$$

Dans le cas d'un graphe orienté, la densité est obtenue en divisant par deux la densité obtenue précédemment. Un *graphe dense* est donc un graphe où le nombre d'arêtes est proche du nombre maximal d'arêtes. Dans le cas contraire où le graphe contient peu d'arêtes. On parle alors de *graphe creux*. Les graphes peuvent être représentés à l'aide de diverses structures de données, notamment les matrices d'adjacence et les listes d'adjacence, que nous présentons ici [57] :

**Matrices d'adjacence** La matrice d'adjacence  $A$  d'un graphe  $G$  est une matrice où chaque élément  $a_{ij}$  de  $A$  correspond au nombre d'arêtes entre les nœuds d'indices  $i$  et  $j$  dans le cas où  $G$  est simple et non pondéré. Si  $G$  est un graphe pondéré, alors chaque élément  $a_{ij}$  correspond au poids associé à l'arête reliant les nœuds  $v_i$  et  $v_j$ . Le terme matrice d'adjacence est utilisé invariablement dans les deux cas. Pour un graphe donné composé de  $n$  nœuds, l'espace nécessaire en mémoire sera de  $n^2$ . La matrice d'adjacence correspondant au graphe simple présenté Figure 16 est donc :

$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \end{matrix}$$

FIGURE 17 – Matrice d'adjacence du graphe simple de la Figure 16.

Les matrices d'adjacence ont des propriétés intéressantes, parmi lesquelles :

- *symétrie* : dans le cas d'un graphe  $G$  simple et non orienté, la matrice d'adjacence  $A$  de  $G$  est symétrique, *i.e*  $A_{ij} = A_{ji}$
- *calcul de la longueur des chemins* : chaque élément  $(i, j)$  de la matrice  $A^n$  correspond au nombre de chemins de longueur  $n$  entre les nœuds d'indices  $i$  et  $j$ .

— ajout et la suppression d'arêtes se font en temps constant,  $O(1)$ <sup>1</sup>

On appelle *matrice des degrés* la matrice diagonale contenant l'information sur le *degré* de chaque nœud, c'est-à-dire le nombre de relations avec d'autres nœuds du graphe. Soit D la matrice des degrés. On a :

$$D_{ij} = \begin{cases} \deg(v_i) & \text{si } i = j \\ 0 & \text{sinon} \end{cases} \quad (19)$$

La matrice des degrés correspondant au graphe d'exemple est donc :

$$D = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \end{matrix}$$

FIGURE 18 – Matrice des degrés du graphe simple de la Figure 16.

Dans le cas où le graphe est pondéré, alors le degré d'un nœud est égal à la somme des poids des relations avec d'autres nœuds.

La représentation sous forme de matrice d'adjacence n'est cependant pas adaptée dans tous les cas d'utilisation, en fonction du type de graphe et de la problématique. En effet, dans le cas d'un graphe creux, une matrice d'adjacence occupera toujours un espace de l'ordre de  $O(n^2)$  en mémoire, bien que très peu d'éléments de la matrice aient une valeur non nulle. Dans ce cas, il peut être plus judicieux d'utiliser une autre représentation : les listes d'adjacence.

**Les listes d'adjacence** Une *liste d'adjacence* contient, pour chaque sommet d'un graphe G, la liste des sommets adjacents à ce sommet. La liste d'adjacence correspondant au graphe présenté Figure 16 est donc :

- 1 : [3]
- 2 : [3,4]
- 3 : [2,5]

---

1. La notation en grand  $O$  est une notation mathématique permettant de décrire le comportement d'une fonction lorsque ses arguments tendent vers une certaine valeur. En informatique, cette notation permet de classer les algorithmes selon leur temps de calcul en fonction de la taille de l'entrée.

- 4 : [2,5]
- 5 : [3,4]

Cette représentation permet l'économie de mémoire dans le cas de graphes creux, dans la mesure où l'espace nécessaire pour la stocker est proportionnel aux nombres d'arêtes dans le graphe. Elle permet par ailleurs d'obtenir la liste de nœuds adjacents à un nœud donné en temps constant, contrairement aux matrices d'adjacence où cette opération se fait en  $O(|V|)$  [58], temps nécessaire pour scanner chaque élément de la ligne correspondante. Cependant, l'utilisation de listes d'adjacence présente aussi certains inconvénients, parmi lesquels [58] :

- La complexité liée à l'ajout/suppression d'arêtes, en  $O(|E|/|V|)$
- La complexité, relativement aux matrices d'adjacence, de vérifier s'il existe une arête entre 2 nœuds en particulier, en  $O(|E|/|V|)$
- Une certaine difficulté à ajouter ou supprimer des arêtes, en  $O(|E|)$

L'utilisation de l'une ou l'autre de ces structures dépend donc du problème à résoudre et du type de graphes considéré. Dans le cas où la tâche principale consiste à rechercher le nombre d'arêtes entre nœuds prédéterminés et où le graphe est dense, il sera préférable d'utiliser une représentation sous forme de matrice d'adjacence. Dans le cas où la tâche principale consiste à retrouver l'ensemble des nœuds liés à un nœud donné, et où le graphe est creux, il sera préférable d'utiliser une représentation sous forme de liste d'adjacence. En pratique cependant, les problèmes ne se distinguent pas de manière aussi nette, et il s'agira de faire les bons compromis en fonction de la situation.

**Parcours et traitement de graphes** On appelle *parcours de graphe* le processus consistant à visiter l'ensemble des nœuds de celui-ci. Deux algorithmes majeurs sont utilisés à cette fin, l'algorithme de *recherche en largeur* (ou *BFS*, Breadth-First Search), et l'algorithme de *recherche en profondeur* (ou *DFS*, Depth-First Search).

**Recherche en largeur** Cet algorithme parcourt le graphe, à partir d'un nœud racine défini, en considérant à chaque étape les voisins des nœuds de l'étape précédente. Le pseudocode de DFS est présenté Algorithme 3.

---

**Algorithm 3** BFS

---

```
1 : procedure BFS(G,s)
2 :   Créer une file vide F
3 :   Ajouter le nœud racine s dans F
4 :   Retirer le premier nœud de F, le marquer comme visité, et le considérer comme nœud
      courant
5 :   Ajouter l'ensemble des voisins du nœud courant dans la file
6 :   Si  $F \neq \{\}$ , reprendre à l'étape 3
7 : Fin procedure
```

---

La Figure 19 présente l'ordre dans lequel les nœuds sont visités par cet algorithme.

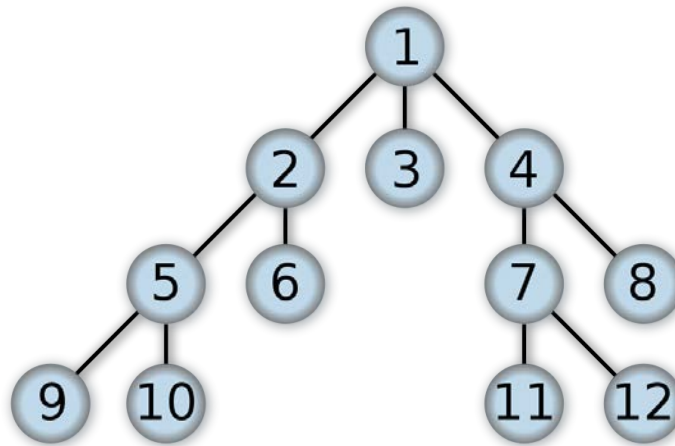


FIGURE 19 – Exemple de parcours en largeur [59].

Dans le pire des cas, chaque nœud et chaque arête est parcouru, la complexité temporelle de l'algorithme de parcours en largeur est donc  $O(|V| + |E|)$ ,  $|V|$  étant le nombre de nœuds et  $|E|$  le nombre d'arêtes dans le graphe [60]. L'espace nécessaire en mémoire pour son exécution varie quant à lui en fonction de la structure utilisée afin de représenter le graphe. Il est proportionnel au nombre de nœuds et d'arêtes dans le cas d'une *liste d'adjacence*, et au carré du nombre de nœuds dans le cas d'une *matrice d'adjacence* [60].

**Recherche en profondeur** L'algorithme de recherche en profondeur parcourt l'ensemble des nœuds du graphe à partir d'un nœud racine  $s$ , en explorant aussi loin que possible chaque branche du graphe avant d'effectuer un "*retour sur trace*" (ou *backtracking*). Le pseudocode présentant le fonctionnement de la recherche en profondeur est présenté algorithme

4.

---

**Algorithm 4** DFS

---

```
1 : procedure DFS(G,s)
2 :   Marquer le nœud s
3 :   Pour chaque nœud n voisin de s Faire
4 :     Si n n'est pas marqué Alors
5 :       DFS(G,n)
6 :     Fin Si
7 :   Fin Pour
8 : Fin procedure
```

---

Il s'agit ici de la version récursive de l'algorithme. Une version itérative est présentée algorithm 5.

---

**Algorithm 5** DFS-Itérative

---

```
1 : procedure DFS-I(G,s)
2 :   Créer une pile vide S
3 :   Empiler s dans S
4 :   Tant que  $S \neq \{\}$  Faire
5 :      $s \leftarrow S.pop()$ 
6 :     Si s n'est pas marqué Alors
7 :       Marquer s
8 :       Pour tout voisin v de s Faire
9 :         Empiler v dans S
10 :      Fin Pour
11 :    Fin Si
12 :  Fin Tant que
13 : Fin procedure
```

---



La Figure 20 présente un exemple de parcours de graphe par l’algorithme de recherche en profondeur. On note la différence dans l’ordre de parcours avec la méthode de recherche en largeur.

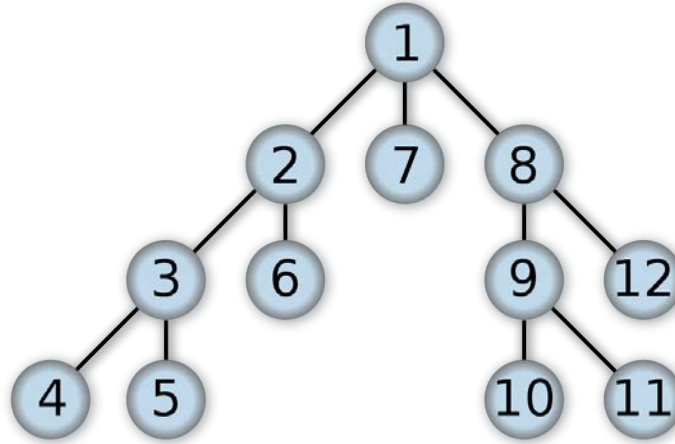


FIGURE 20 – Exemple de parcours en profondeur [61].

Au sein de ces deux exemples, le graphe est un arbre : le cas où des cycles existent est donc éliminé. Cependant, les algorithmes de recherche en largeur et en profondeur sont tout aussi adaptés aux graphes, grâce au marquage de nœuds visités. Tout comme l’algorithme de parcours en largeur, la complexité temporelle de l’algorithme de parcours en profondeur est en  $O(|V| + |E|)$ . Ces deux algorithmes permettent, entre autres, d’obtenir l’ensemble des chemins entre deux nœuds du graphe. Cependant, ces approches exhaustives peuvent ne pas être adaptées dans le cas de grands graphes, en particulier si l’on s’intéresse seulement aux chemins entre nœuds spécifiques. Il existe des algorithmes de parcours de graphe s’appuyant sur des *heuristiques*, tel que *best-first search*, *Dijkstra* [62] ou encore  $A^*$  [63], évitant de parcourir l’ensemble du graphe et réduisant ainsi l’espace de recherche. Ces heuristiques sont définies en fonction des données et du problème à résoudre. Elles peuvent s’appuyer sur des mesures de distance ou de similarité entre nœuds : distance euclidienne, distance de Manhattan, similarité cosinus, indice de Jaccard, etc.

### 2.5.2 Application des méthodes de fouilles aux graphes : notion de noyaux de graphes

Soit deux graphes  $G$  et  $G'$  deux graphes appartenant à l'espace des graphes  $\mathcal{G}$ . Afin d'effectuer une comparaison de ces deux graphes, il est nécessaire de trouver une fonction  $f : \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}$ . Cette fonction  $f$  quantifie une mesure de similarité entre deux graphes. On retrouve ici la notion de noyau vue au chapitre 2.4.3. Il s'agit ici de noyaux appliqués aux graphes, ou *Graph Kernel*.

De nombreux noyaux ont été proposés dans la littérature, le plus souvent basés sur la notion de décompte de sous-structures communes au sein des graphes. Ces sous-structures peuvent être obtenues *via* le parcours en largeur ou en profondeur du graphe à partir d'un noeud racine. L'un des noyaux de référence est le noyau de Weisfeiler-Lehman. Ce dernier est basé sur la notion de réétiquetage de chaque noeud d'un graphe grâce à une fonction d'étiquetage  $l$ . A chaque itération  $i$  de l'algorithme, une nouvelle étiquette  $l_i(v)$  est obtenue pour chaque noeud  $v$ . Soit  $w$  la fonction réétiquettant un graphe  $G$ , telle que  $w(V, E, l_i) = (V, E, l_{i+1})$ .

On définit une séquence de graphes de Weisfeiler-Lehman de profondeur  $h$   $\{G_0, \dots, G_h\}$  comme la séquence  $\{(V, E, l_0), (V, E, l_1), \dots, (V, E, l_h)\}$ . La définition générale du noyau de Weisfeiler-Lehman est la suivante :

$$k_{WL}^h(G, G') = \alpha_0 k(G_0, G'_0) + \alpha_1 k(G_1, G'_1) + \dots + \alpha_h k(G_h, G'_h) \quad (20)$$

Une des variantes de ce noyau est le *subtree kernel* de Weisfeiler-Lehman. Soit  $\Sigma_i \subseteq \Sigma$  l'ensemble des lettres apparaissant dans les étiquettes des noeuds au moins une fois à l'issue de la  $i^{me}$  itération,  $c_i(G, \sigma_{ij})$  le nombre d'occurrences de la lettre  $\sigma_{ij}$  dans le graphe  $G$ . On définit le *Weisfeiler-Lehman subtree kernel*  $k_{WLsubtree}$  comme :

$$k_{WLsubtree}^h(G, G') = \langle \Phi_{WLsubtree}^h(G), \Phi_{WLsubtree}^h(G') \rangle \quad (21)$$

, avec  $\Phi_{WLsubtree}^{(h)}(G) = (c_0(G, \sigma_{01}), \dots, c_0(G, \sigma_{0|\Sigma_0|}), \dots, c_h(G, \sigma_{h1}), \dots, c_h(G, \sigma_{h|\Sigma_h|}))$ . Les figures suivantes permettent de détailler dans un cas pratique le calcul de ce noyau.

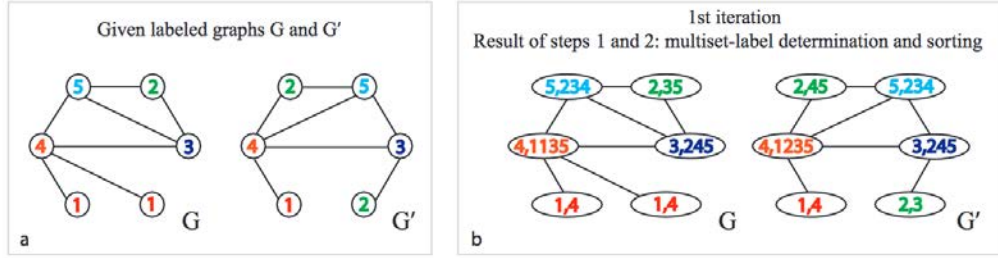


FIGURE 21 – (a) soient deux graphes G et G', dont on cherche à calculer le noyau. La première étape ([b]) de l'algorithme consiste à réétiqueter les noeuds en fonction de leurs voisins directs. Un tri est effectué, afin que l'ordre reste le même pour tous les noeuds. Par exemple, le noeud le plus à gauche, étiqueté 4, est voisin de 4 noeuds, étiquetés  $\{1, 1, 3, 5\}$ . Lors de l'étape de réétiquetage, ce noeud prend ainsi l'étiquette 4, 1, 1, 3, 5 [64].

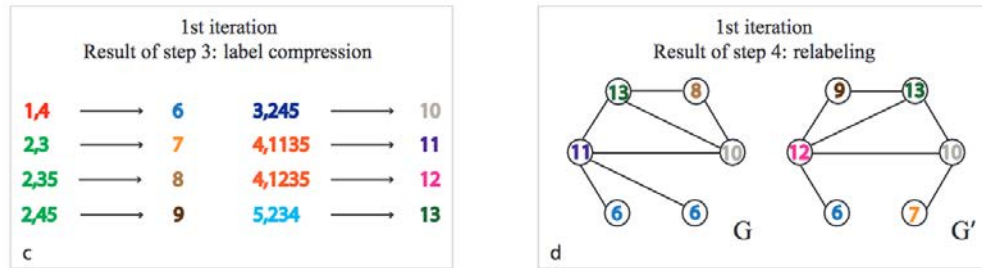


FIGURE 22 – (c) L'étape suivante consiste à compresser les nouvelles étiquettes, en agrandissant l'alphabet de nouvelles lettres. (d) On attribue enfin les nouvelles étiquettes aux noeuds correspondants. On constate que ces dernières donnent une information sur le voisinage direct des noeuds à l'étape précédente : le noeud le plus à droite, ayant le même voisinage à l'étape initiale, obtient la même étiquette 10 à la fin de la première itération [64].

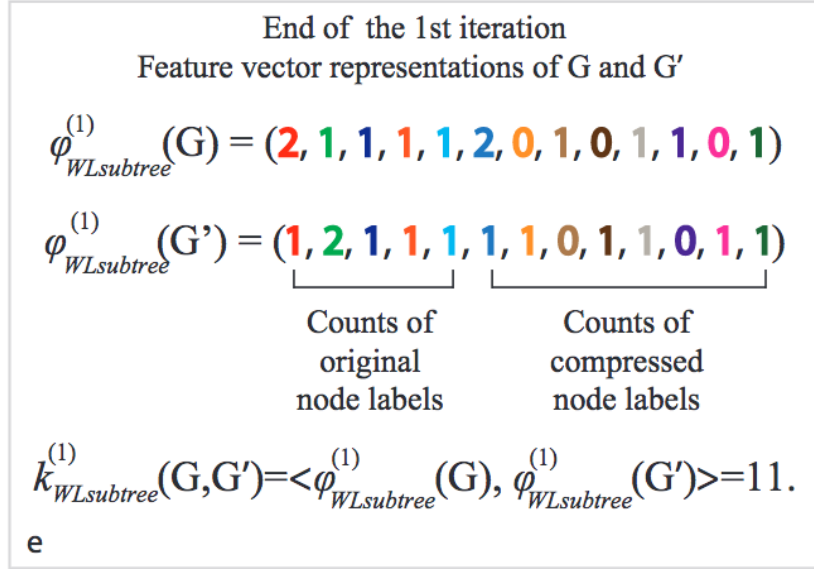


FIGURE 23 – La dernière étape est celle de la construction des vecteurs représentant chaque graphe. Chacune des composantes de ces derniers correspond au décompte de chaque étiquette dans les graphes, à chacune des itérations. La taille du vecteur dépend donc du nombre d'étiquettes distinctes dans le graphe à la fin des itérations, et du nombre d'itérations [64].

Le calcul de noyaux sur les graphes permet de faire un pont entre les méthodes d'apprentissage sur les structures et les méthodes d'apprentissage basées sur les attributs, et permet l'application de méthodes d'Apprentissage supervisé et non supervisé sur les graphes. Cependant, la définition d'un bon noyau, efficacement calculable et limitant les pertes d'informations, peut-être délicate. Ces noyaux sont souvent retrouvés dans la littérature, notamment en chemoinformatique ou en biologie moléculaire.

## 2.6 Évaluation des tâches d'apprentissage

Nous avons vu au cours des chapitres précédents quelques exemples de méthodes d'Apprentissage supervisé. Afin de pouvoir évaluer la qualité du modèle obtenu à la suite de l'apprentissage, il est nécessaire de mettre en place des métriques objectives.

Dans le cadre de la classification, la réalisation d'une matrice de confusion (ou table de contingence) est souvent la première étape afin de calculer des métriques plus complexes. Cette matrice est constituée par :

- Les vrais positifs (VP), les éléments affectés à la classe  $y$  et appartenant effectivement à cette classe,
- Les vrais négatifs (VN), les éléments affectés à la classe  $\bar{y}$  et appartenant effectivement à cette classe,
- Les faux positifs (FP), les éléments affectés à la classe  $y$  et appartenant à une autre classe,
- Les faux négatifs, les éléments affectés à la classe  $\bar{y}$  et appartenant à une autre classe.

Le placement de ces éléments dans la table de contingence est présenté Table 2.

TABLE 2 – Exemple de matrice de confusion

	Prédiction +	Prédiction -
Classe réelle +	TP	FN
Classe réelle -	FP	TN

On retrouve dans la littérature les notions *d'erreur de type 1* et *d'erreur de type 2*. Elles correspondent aux faux positifs et aux faux négatifs, respectivement.

De nombreuses métriques existent afin d'évaluer la qualité d'un classifieur. Nous allons ici présenter quelques métriques que sont la Précision, le Rappel et la F-mesure.

### La précision

La Précision mesure, parmi tous les éléments prédits comme appartenant à une classe, la proportion appartenant effectivement à cette classe. En d'autres termes, elle mesure la pertinence de la classification. Plus formellement :

$$Précision : \frac{TP}{TP + FP} \quad (22)$$

### Le rappel

Le Rappel est une mesure complémentaire à la précision. Son calcul est présenté Équation 23.

$$Rappel : \frac{TP}{TP + FN} \quad (23)$$

Le Rappel quantifie la proportion d'éléments prédits comme appartenant à une classe parmi tous les éléments appartenant effectivement à cette classe.

### La F-mesure

Les deux mesures précédentes constituent de bons indicateurs concernant la classification. Cependant, pris indépendamment, elles ne suffisent pas pour conclure sur la qualité d'un modèle. En effet, il est possible pour un classifieur d'avoir un très bon Rappel, *i.e.*, de classer un maximum d'éléments *réellement* positifs en tant que positifs, en classant l'ensemble des objets comme positifs. Ce faisant, le Rappel obtenu est de 1. Cependant, la Précision sera fortement réduite.

La F-mesure est une **mesure composite**, combinant Précision et Rappel. Elle est définie par la relation :

$$\frac{2 \times \text{précision} \times \text{rappel}}{\text{précision} + \text{rappel}} \quad (24)$$

Il s'agit de la moyenne harmonique des deux métriques vues précédemment. Sa valeur va de 0 pour un mauvais classifieur, à 1, pour une classification parfaite.

### La Kappa statistique

La kappa statistique est une métrique permettant de chiffrer l'intensité de l'accord réel entre des jugements. Elle est notamment utilisée dans les méthodes d'ensemble telles que les forêts d'arbres aléatoires, afin de quantifier l'accord entre les différents arbres générés. Landis *et al.* [65] ont proposé une interprétation de cette mesure, présentée Table suivante.

$\kappa$	Interprétation
inf 0	Désaccord
0.0–0.20	Accord très faible
0.21–0.40	Accord faible
0.41–0.60	Accord modéré
0.61–0.80	Accord fort
0.81–0.99	Accord presque parfait
1.00	Accord parfait

## 3 Le Web des données

### 3.1 Web des données, Web sémantique, Données ouvertes et liées : définitions

Le Web des données, défini par Sir Tim Berners-Lee et porté par le W3C, est une initiative visant à favoriser la publication de données structurées sur le Web, non pas sous la forme de données isolées les unes des autres, mais reliées entre elles afin de constituer un réseau global d'informations [66].

L'exploitation des données par les machines est au cœur de l'initiative. En effet, elle vise, en s'appuyant sur les standards du Web préexistants (HTTP, URI, etc.), à faciliter le partage et la recherche d'informations entre machines, leur permettant d'interroger les données présentes sur le Web, d'effectuer un lien entre elles et de leur associer une sémantique.

On retrouve cette notion dans «Weaving the Web — The original design and ultimate destiny of the World Wide Web, by its inventor» [67] : *”The first step is putting data on the Web in a form that machines can naturally understand, or converting it to that form. This creates what I call a Semantic Web – a web of data that can be processed directly or indirectly by machines.»*

Quatre principes ont été définis par Tim Berners Lee :

- identifier les objets par des adresses uniques, les **URI**
- ajouter des URI externes aux données pour améliorer la découverte d'autres informations sur le Web
- utiliser des adresses URI qui existent sur le Web
- fournir *via* l'URI des renseignements lisibles par les humains, mais aussi par les machines (par exemple pour afficher un page au format HTML à un humain, et une page au format XML— RDF pour une machine).

Les données liées (*Linked Data*) sont les données, informations et connaissances effectivement connectées et partagées au sein du Web sémantique, grâce aux URI et au RDF, un langage

de description des données, que nous verrons dans la section suivante. Le terme de **Web sémantique** est parfois retrouvé, et source de confusion. Tim-Berners Lee décrit les données ouvertes et liées comme "*The semantic Web done right*" [68].

Les données ouvertes et liées (*Linked Open Data* ou LOD en anglais) constituent un ensemble de données structurées, disponibles sur le Web et interconnectées entre elles [66]. Ces données sont généralement issues de sources de données initialement indépendantes, *i.e.*, non connectées et difficilement accessibles à des outils logiciels. La communauté du Web sémantique, soutenue par le W3C (World Wide Web Consortium), encourage les propriétaires de données à (i) transformer leurs données dans le format standard des données liées qu'est RDF, (ii) les publier en ligne et (iii) définir un maximum de liens entre leurs données et les LOD déjà publiées [69]. Il résulte de ce processus un ensemble de sources de données connectées entre elles et librement accessibles, ce qui offre des possibilités d'utilisation inédites. Dans le domaine biomédical, le contenu de nombreuses bases de données biologiques publiques (comme OMIM, UniProt, DrugBank, etc.) est disponible en RDF via les portails Web d'initiatives comme le projet Bio2RDF ou la plateforme RDF de l'EBI (European Bioinformatics Institute) [70][71]. La Figure 24 est une cartographie des sources de données liées comme elles étaient référencées en avril 2014 par l'initiative lod-cloud.net. Des données provenant d'une grande variété de domaines sont présentes au sein de cette représentation : données linguistiques, biomédicales, géographiques, etc. Dans cette thèse, nous nous intéresserons particulièrement aux données biomédicales, présentes dans la partie inférieure droite du graphe (voir Figure 25)

## 3.2 Un ensemble de standards

### 3.2.1 *One entity to rule them all* : le W3C

Le W3C, ou *World Wide Web Consortium*, se charge de la standardisation des outils et protocoles associés au Web. Cette organisation a été créée en octobre 1994 par Tim-Berners Lee au MIT, et réunit des membres de plusieurs entreprises et organisations, dont la mission est de maintenir et de développer les standards utilisés dans le monde entier. Parmi ces multiples missions, le W3C définit et maintient à jour les standards, outils et protocoles





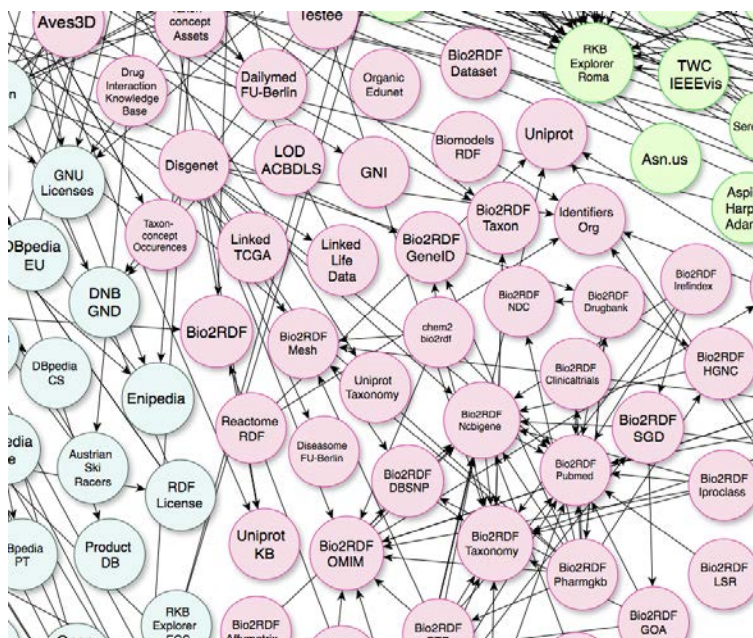


FIGURE 25 – Zoom sur les données liées du monde médical. Source : lod-cloud.net API

- A la suite de ces éléments, l'on retrouve le nom de l'hôte. Dans le cas d'une **URL** (*Uniform Resource Locator*), identifiant une ressource accessible sur le réseau, l'hôte correspond à l'adresse IP de la machine. Dans le cas plus général d'un URI, la notion d'accessibilité et de localisation n'a pas toujours un sens.
- Un chemin, permettant de spécifier la ressource.
- Un fragment, qui permet de faire référence à une ressource dérivant d'une autre ressource.

Ainsi, <http://bio2rdf.org/ncbigene#37191> est, par exemple, l'URI associé au gène codant pour le récepteur 5HT-1B. Ici, [bio2rdf.org/ncbigene](http://bio2rdf.org/ncbigene) indique que la ressource provient de la ressource Bio2RDF et est associée à l'identifiant de gène 37191. On voit ici l'intérêt du fragment : le préfixe restera le même pour toutes les entités de type gène [bio2rdf.org/ncbigene](http://bio2rdf.org/ncbigene), uniquement l'identifiant changera entre deux gènes différents.

Bien que dans ce cas, la sémantique de l'URI soit claire et permet une certaine homogénéité de la codification, il n'existe pas d'obligation à ce que ce soit le cas. La tâche incombe aux producteurs et consommateurs d'URI de se coordonner sur la définition et l'utilisation des identificateurs. Certaines initiatives, telles que le Dublin Core Schema [73], visent cependant

à fournir un schéma et un vocabulaire générique à la disposition des différents acteurs.

### 3.2.3 Un modèle standard de représentation de données : RDF

Le **RDF**, pour **Resource Description Framework**, est un modèle standardisé pour l'échange des données sur le Web. Les documents structurés dans le format RDF permettent de décrire des relations entre entités sous la forme d'ensemble de triplets. Un triplet permet de représenter un lien, sous la forme sujet — prédicat — objet. Le sujet correspond à une ressource, et le prédicat décrit une relation entre le sujet et l'objet. Par exemple, «Chien» — «est un» — «Mammifère» pourrait être un triplet représentant l'appartenance des chiens au taxon des mammifères. Dans ce format, chaque élément est représenté par une URI, qu'il s'agisse d'un sujet, d'un prédicat ou d'un objet, facilitant l'interopérabilité et la compréhension par tous les acteurs, humains ou non, des relations entre entités.

Plusieurs syntaxes existent et peuvent être utilisées afin d'écrire des documents au format RDF : **RDF-XML** – le plus ancien, Turtle, JSON-LD, etc. Quelle que soit sa syntaxe, le document RDF décrit la même chose, à savoir un graphe dirigé étiqueté. Un exemple de graphe RDF est présenté ci-dessous, Figure 26. Le fichier au format RDF-XML correspondant à ce graphe est présenté Figure 27.

Afin de stocker et manipuler les graphes RDF, des systèmes de gestion spécifiques existent, les *triple stores*, ou bases de triplets. Cependant, il est nécessaire d'avoir à sa disposition un langage de requêtes pour interroger ces données d'intérêt.

### 3.2.4 Un langage de requêtes : SPARQL

SPARQL, acronyme récursif pour **SPARQL Protocol And RDF Query Language**, est un langage de requêtes et un protocole associé permettant d'effectuer des requêtes et de récupérer des données contenues dans des graphes RDF. Les requêtes écrites en SPARQL sont constituées de  **patrons de triplets** , de **conjonctions** et de **disjonctions**. Cette liste d'opérations n'est cependant pas exhaustive – on retrouvera par exemple la notion d'optionnalité et de filtre.

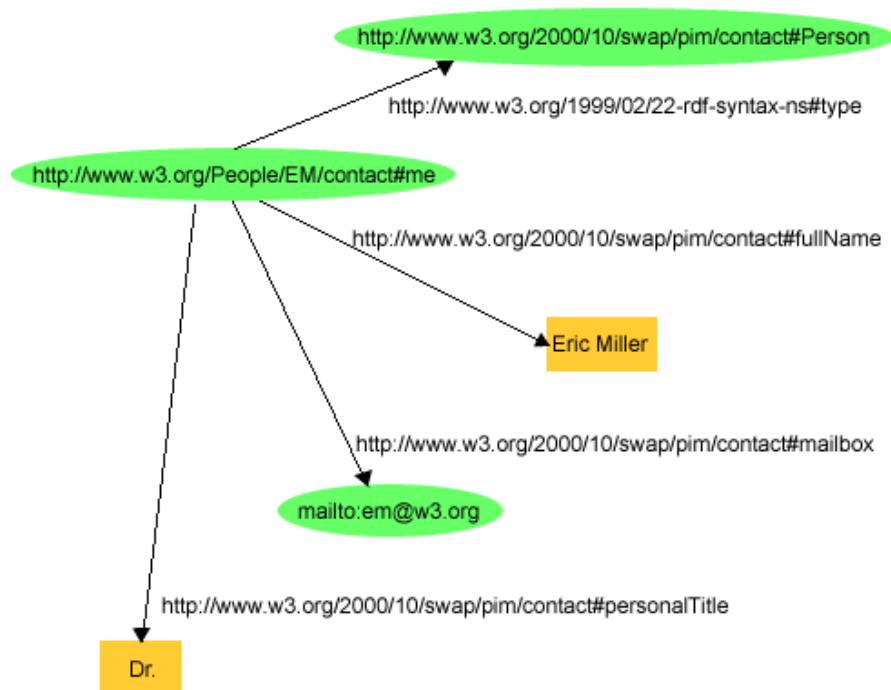


FIGURE 26 – Exemple de graphe RDF représentant l’entité «Eric Miller» [74]. On notera ici que les relations, représentées par des URI, peuvent pointer soit vers des chaînes de caractères (le *titre* d’Eric Miller par exemple), soit vers d’autres URI (par exemple pour le type, une personne. On parle de propriété à type de données (ou *datatype property*, lorsqu’elles pointent vers un valeur) et de propriété à objet (ou *object property*), respectivement.

Quatre types de requêtes peuvent être effectuées :

- les requêtes avec la clause **SELECT**, permettant de sélectionner des données d’un graphe et de les retourner sous forme de tableau,
- les requêtes **CONSTRUCT**, permettant aussi de sélectionner les données correspondantes d’un graphe RDF, mais cette fois en retournant les résultats sous la forme de RDF valide,
- Les requêtes **ASK**, retournant un booléen,
- Les requêtes **DESCRIBE**, retournant des informations à propos des ressources.

Afin de pouvoir rendre accessibles les données à n’importe quel utilisateur, n’importe quel utilisateur peut mettre en place un **point d’accès** SPARQL, permettant d’effectuer des requêtes à partir d’une interface dédiée, ou via un service Web. Un exemple de point d’accès est celui que nous avons mis en place dans le cadre du projet associé à cette thèse. Il est

```

<?xml version="1.0"?>
<rdf :RDF xmlns :rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
          xmlns :contact="http://www.w3.org/2000/10/swap/pim/
contact#">
  <contact :Person rdf :about="http://www.w3.org/People/EM/contact#me">
    <contact :fullName>Eric Miller</contact :fullName>
    <contact :mailbox rdf :resource="mailto:em@w3.org"/>
    <contact :personalTitle>Dr.</contact :personalTitle>
  </contact :Person>
</rdf :RDF>

```

FIGURE 27 – Exemple de graphe représenté au format RDF-XML.

accessible à l'adresse <https://pgxlod.loria.fr>. Son accès est restreint, mais un compte peut être créé sur demande.

## 4 Positionnement

Nous proposons dans cette thèse de (i) sélectionner un sous-ensemble adéquat de LOD et de le compléter pour constituer un jeu de données recouvrant les trois axes de la Pharmacogénomique : les gènes, les maladies et les médicaments ; (ii) transformer et prétraiter ces données afin de pouvoir appliquer des méthodes de fouille, dans le but de (iii) suggérer des relations gène-médicament. Notre travail s'est appuyé sur le projet Bio2RDF [75], projet open source qui vise à utiliser les technologies du Web sémantique afin de lier les connaissances du domaine biologique et biomédical. Ils proposent un ensemble de conventions de nommage et de structuration des données en RDF ainsi une API, permettant à chacun de participer au projet et d'y ajouter un jeu de données. En juillet 2014, déjà plus de 11 milliards de triplets appartenant à 35 jeux de données étaient référencés au sein de Bio2RDF. Afin de mener à bien cette thèse, nous nous sommes inspirés d'un travail similaire réalisé en 2012 par une équipe de l'Université Stanford [76]. Leur objectif était de retrouver de manière computationnelle des interactions médicamenteuses, en exploitant les données déjà présentes dans la littérature scientifique. Leur mode opératoire est le suivant :

- Extraire les relations gènes-médicaments des résumés provenant de Medline, ainsi que le type de relation entre ces derniers («inhibe», «active», etc.). Cette extraction est basée sur le travail précédent de Coulet et *al.* [77].



- Normaliser les relations, *via* l'utilisation d'une ontologie/
- Inférer les relations entre médicaments à partir de paires de relations gènes-médicaments/

Un exemple de relation indirecte entre deux médicaments est présenté Figure 28.

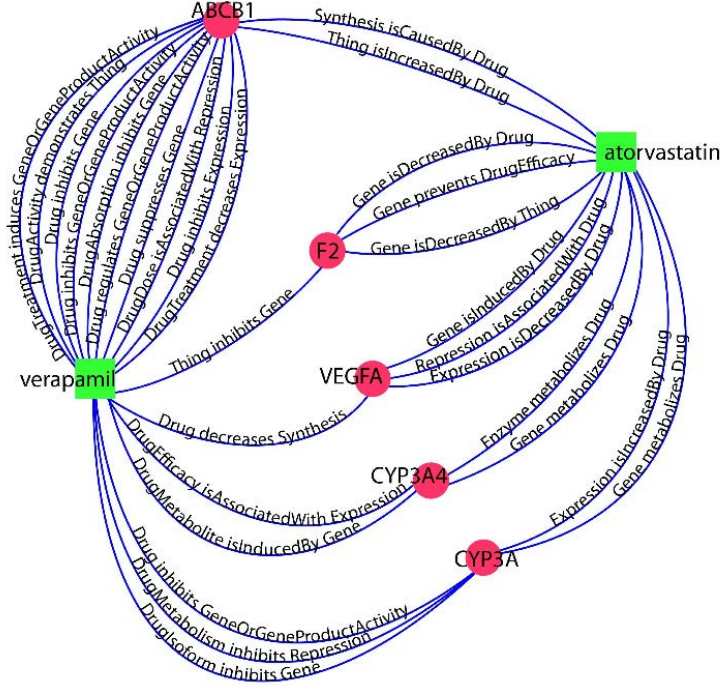


FIGURE 28 – Représentation graphique des liaisons indirectes entre médicaments passant par des gènes, extraites à partir de résumés Medline [78]

À partir de ces relations et d'un ensemble de couples de médicaments dont l'interaction est avérée, l'équipe a entraîné un modèle, en utilisant l'algorithme de la forêt d'arbres aléatoires. L'échantillon d'entraînement était constitué de 5000 témoins positifs, extraits d'une liste d'interactions connues sur DrugBank, et de 5000 témoins négatifs. Les témoins négatifs ont été obtenus en récupérant aléatoirement des paires des médicaments n'apparaissant pas dans la liste de paires connues indexées par DrugBank. Sur le graphe de la Figure 28, chaque chemin entre deux médicaments dont l'interaction est avérée est considéré comme un événement positif, et vice-versa pour un chemin entre deux médicaments appartenant à l'ensemble des témoins négatifs.

Une tâche courante en Pharmacogénomique consiste à identifier des gènes pouvant être impliqués dans la variabilité d'une réponse à un médicament, *i.e*, les pharmacogènes. Cette

tâche étant très chronophage, des approches computationnelles de suggestion de pharmacogènes seraient intéressantes afin de guider les équipes dans leurs recherches. Nous proposons dans ce travail de sélectionner diverses sources de données biomédicales d'intérêt en Pharmacogénomique, et d'y appliquer des méthodes de fouille afin d'identifier et de proposer des pharmacogènes valides. Nous avons fait le choix de nous focaliser sur les données disponibles de manière liée, et, dans le cas où elles ne l'étaient pas, de les transformer. L'utilisation des données ouvertes et liées rend possible l'utilisation conjointe de multiples sources, étant construites et distribuées selon un *framework* commun. Notre problème peut être vu comme un problème de découverte de liens. De nombreux travaux proposent des approches de résolution, *via* l'apprentissage automatisé [79, 80], la Fouille de graphes [81, 82, 83], ou encore par des approches de visualisation [84]. Certaines de ces approches donnent de bons résultats, mais sont fortement dépendantes des données. Dans cette thèse, nous proposons d'utiliser deux méthodes afin de classer les paires gènes-médicaments : les forêts d'arbres aléatoires et les noyaux de graphes, dont les résultats sont comparés.

## 5 Contributions

Les résultats des travaux présentés dans ce document ont donné lieu à une première publication dans les actes de la conférence *Semantic Web for Life Sciences (SWAT4LS 2015)* [85]. J'ai eu le privilège de présenter ce travail sous forme d'une présentation longue à Cambridge (Royaume-Uni). Suite à cette conférence, le travail a été enrichi de nouvelles contributions, soumis et publié au *Journal of Biomedical Semantics (JBMS - BioMed Central)*. Le chapitre suivant reprend l'article publié dans ce journal [86].

Troisième partie

# Suggestion de pharmacogènes valides par apprentissage à partir de données ouvertes et liées



RESEARCH

Open Access



# Learning from biomedical linked data to suggest valid pharmacogenes

Kevin Dalleau<sup>1†</sup>, Yassine Marzougui<sup>1,2†</sup>, Sébastien Da Silva<sup>1</sup>, Patrice Ringot<sup>1</sup>, Ndeye Coumba Ndiaye<sup>3</sup> and Adrien Coulet<sup>1\*</sup> 

## Abstract

**Background:** A standard task in pharmacogenomics research is identifying genes that may be involved in drug response variability, i.e., pharmacogenes. Because genomic experiments tended to generate many false positives, computational approaches based on the use of background knowledge have been proposed. Until now, only molecular networks or the biomedical literature were used, whereas many other resources are available.

**Method:** We propose here to consume a diverse and larger set of resources using linked data related either to genes, drugs or diseases. One of the advantages of linked data is that they are built on a standard framework that facilitates the joint use of various sources, and thus facilitates considering features of various origins. We propose a selection and linkage of data sources relevant to pharmacogenomics, including for example DisGeNET and Clinvar. We use machine learning to identify and prioritize pharmacogenes that are the most probably valid, considering the selected linked data. This identification relies on the classification of gene–drug pairs as either pharmacogenomically associated or not and was experimented with two machine learning methods –random forest and graph kernel–, which results are compared in this article.

**Results:** We assembled a set of linked data relative to pharmacogenomics, of 2,610,793 triples, coming from six distinct resources. Learning from these data, random forest enables identifying valid pharmacogenes with a F-measure of 0.73, on a 10 folds cross-validation, whereas graph kernel achieves a F-measure of 0.81. A list of top candidates proposed by both approaches is provided and their obtention is discussed.

**Keywords:** Linked data, Pharmacogenomics, Data mining, Knowledge discovery from databases, Machine learning, Valid pharmacogenes

## Background

Pharmacogenomics (PGx) studies how individual gene variations cause variability in drug responses [1]. Well established knowledge in PGx constitutes a basis for implementing personalized medicine, i.e., a medicine tailored to each patient by considering in particular her/his genomic context. The state of the art of this domain lies both in the biomedical literature and in specialized databases [2, 3], but a large part of it is controversial, and not yet applicable to medicine. Indeed, this results from studies difficult to reproduce and that do not fulfill statistical validation standards for two main reasons: the

small size of populations involved in studies because of the rarity of gene variants studied and the potential coaction of several variants [4, 5]. It is consequently of interest to the PGx community to explore any source of evidence that may contribute to confirming or moderating PGx state of the art. So far, existing works used either molecular network databases or the biomedical literature (see “Discovery of pharmacogenes” subsection). We propose in this work to explore how other resources, and particularly Linked Open Data (LOD) may be useful in this domain.

## Linked open data

LOD are constituting a large and growing collection of datasets that present the main advantages of being represented in a standard format (based on both RDF and URIs) and partially connected to each other and to domain

\*Correspondence: adrien.coulet@loria.fr

†Equal contributors

<sup>1</sup>LORIA (CNRS, Inria Nancy-Grand Est, University of Lorraine), Campus Scientifique, Nancy, France

Full list of author information is available at the end of the article

knowledge represented within semantic web ontologies [6]. For these reasons, LOD offer novel opportunities for the development of successful data integration and knowledge discovery campaign, as required for the discovery of novel pharmacogenes. LOD are part of a community effort to build a semantic web, where web and data resources can be interpreted both by humans and machines. The recent availability of LOD is particularly beneficial to the life sciences, where relevant data are spread over various data sources with no agreement on a unique representation of biological entities [7]. Consequently, data integration is an initial challenge one faces if one wants to mine life science data considering several data sources. Various initiatives such as Bio2RDF, the EBI platform, PDBj and Linked Open Drug Data (LODD) aim at pushing life sciences data into the LOD cloud with the idea of facilitating their integration [8–11]. It results from these initiatives a large collection of life-science data, unequally connected but in a standard format and available for mining. Despite good will and emerging standard practices for publishing data as LOD, several drawbacks make their use still challenging [12, 13]. Among existing difficulties we can cite the limited amount of links between datasets and the limits of implementations of federated queries.

#### Pharmacogenomics data and linked data

PharmGKB is a comprehensive database about PGx that includes manually annotated gene–drug relationships [3]. Recently, annotations of PharmGKB have been completed with a *level of evidence* going from 1 to 4, distinguishing well validated gene–drug relationships (level= 1–2) from insufficiently validated ones (3–4), thus pointing at knowledge in need for additional investigations [14]. PharmGKB does not provide its data in RDF, but parts of PharmGKB have been transformed and published in RDF by contributors of the Bio2RDF project, thus enabling SPARQL queries [15]. Clinical annotations of PharmGKB are however not freely available. Their usage is granted through a license agreement, preventing the data from being redistributed, thus published as Linked Open Data. Many other databases provides data that are indirectly relevant to PGx. For instances, DrugBank [16] provides drug–target relationships; ClinVar [17] provides gene variant–phenotype relationships; SIDER [18, 19] and Medi-Span provides drug–phenotype relationships such as drug adverse events or indications [20]. Medi-Span is a proprietary database of Wolters Kluwer Health (Indianapolis, IN) aiming at providing drug clinical data to clinicians. DGIdb (The Drug Gene Interaction database) is another interesting initiative that integrates quasi-exhaustively data about gene–drug relationships, considering 15 distinct sources [21]. DisGenet is a data integration initiative that focuses on gene–disease

relationships and provides data in RDF, including parts of ClinVar and OMIM [22].

Data integration effort clearly oriented to PGx applications are less common, particularly if considering semantic web approaches [23]. Hoehndorf et al. integrated and made available a set of PGx related data that includes PharmGKB, DrugBank and CTD (the Comparative Toxicogenomics Database), using semantic web technologies [24]. They used the integrated dataset to identify pathways that may be perturbed in PGx. In this effort of publishing PGx data, Coulet et al. extracted about 40,000 PGx relationships from the biomedical literature and published them in the form of RDF statements [25].

#### Mining linked data

Suggesting valid pharmacogenes in this work is seen as proposing novel gene–drug relationships from an RDF graph, which in turn can be described as a link prediction problem. Many works have focused on the link prediction problem, studying various approaches such as machine learning [26, 27], graph mining [28–30], identity resolution [31, 32] and data visualisation [33]. Some of these methods obtain good results, but all are dependent from the input graphs (its quality, topology, etc.) and are hard to reuse for new applications. Recently, de Vries and de Rooij proposed a complete framework for applying Graph Kernel (GK) in an adaptive manner to RDF graphs [34]. GK are machine learning methods that have the ability to deal directly with graph data, particularly by computing kernel functions that evaluate similarity between graphs or pieces of graphs [35]. The framework of de Vries and de Rooij is implemented in an open source library named *Mustard* [36]. It enables classifying RDF instances considering their neighborhood in the graph. This neighborhood is encoded within features such as labels of edges or graph substructures such as *walks* (i.e., linear paths) or *sub-graphs*. In the work we present here, we reused Mustard and fitted its capability of instance classification to the case of link prediction.

In relation with PGx research, Percha et al. mined the set of RDF statements extracted from text by Coulet et al. with a Random Forest (RF) algorithm and successfully predicted drug–drug interactions [37]. With the aim of predicting pharmacogenes, we experimented as Percha et al. with the RF algorithm in the preliminary stage of this work [38]. First results we obtained with RF are here updated and compared with GK approaches.

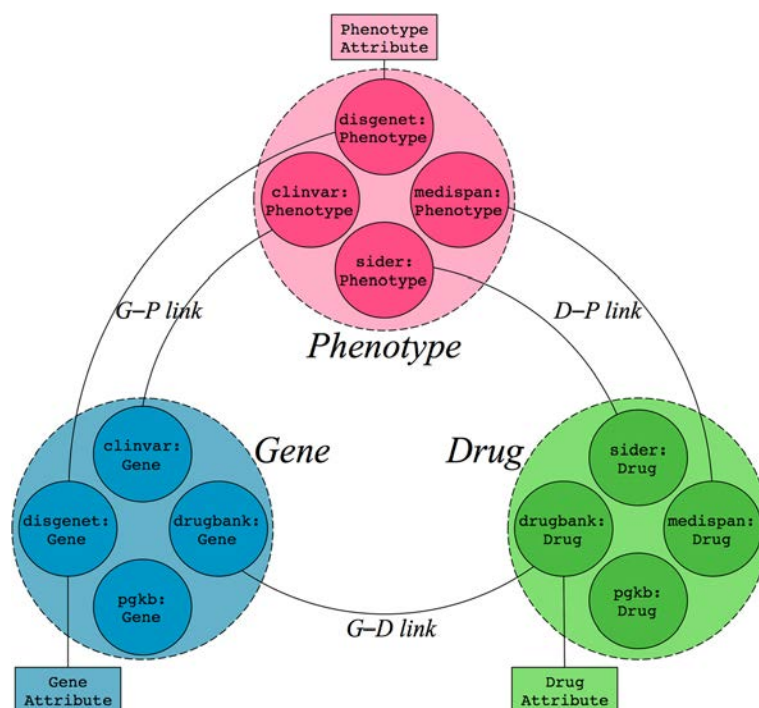
#### Discovery of pharmacogenes

Hansen et al. proposed a method based on a logistic classifier to generate candidate pharmacogenes, using data from PharmGKB, DrugBank, and protein–protein interactions from InWeb [39]. An issue with this approach is that PharmGKB and DrugBank are manually curated

Semantic web technologies have also been experimented for PGx knowledge discovery. Dumontier and Villanueva-Rosales proposed a knowledge representation of the domain and benefit from reasoning mechanisms to answer sophisticated queries related to depression drugs [42]. Coulet et al. used patient data to instantiate a description logics knowledge base, then extracted association rules from it to identify *gene variant–drug response* associations [43]. More generally, advantages that semantic web technologies may offer to PGx and personalized medicine are listed in [23].

## Methods

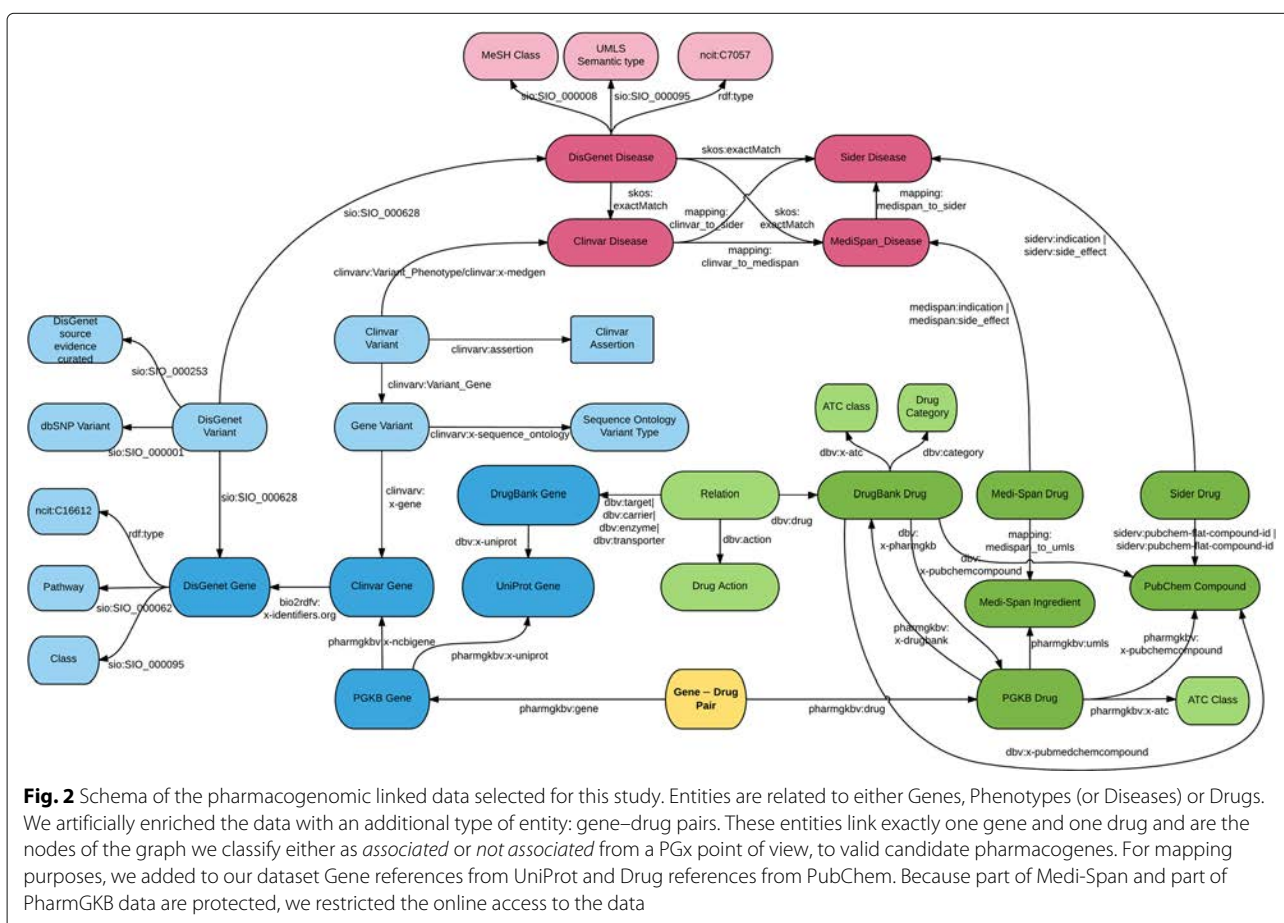
**Data selection** Initial step is to select a set of data that include relevant data about PGx gene–drug relationships. Figure 1 gives a general overview of the type of data we consider for this study: three types of entities, *gene*, *phenotype* and *drug*; and relationships between them, i.e., *gene–phenotype*, *phenotype–drug* and *gene–drug* relationships.

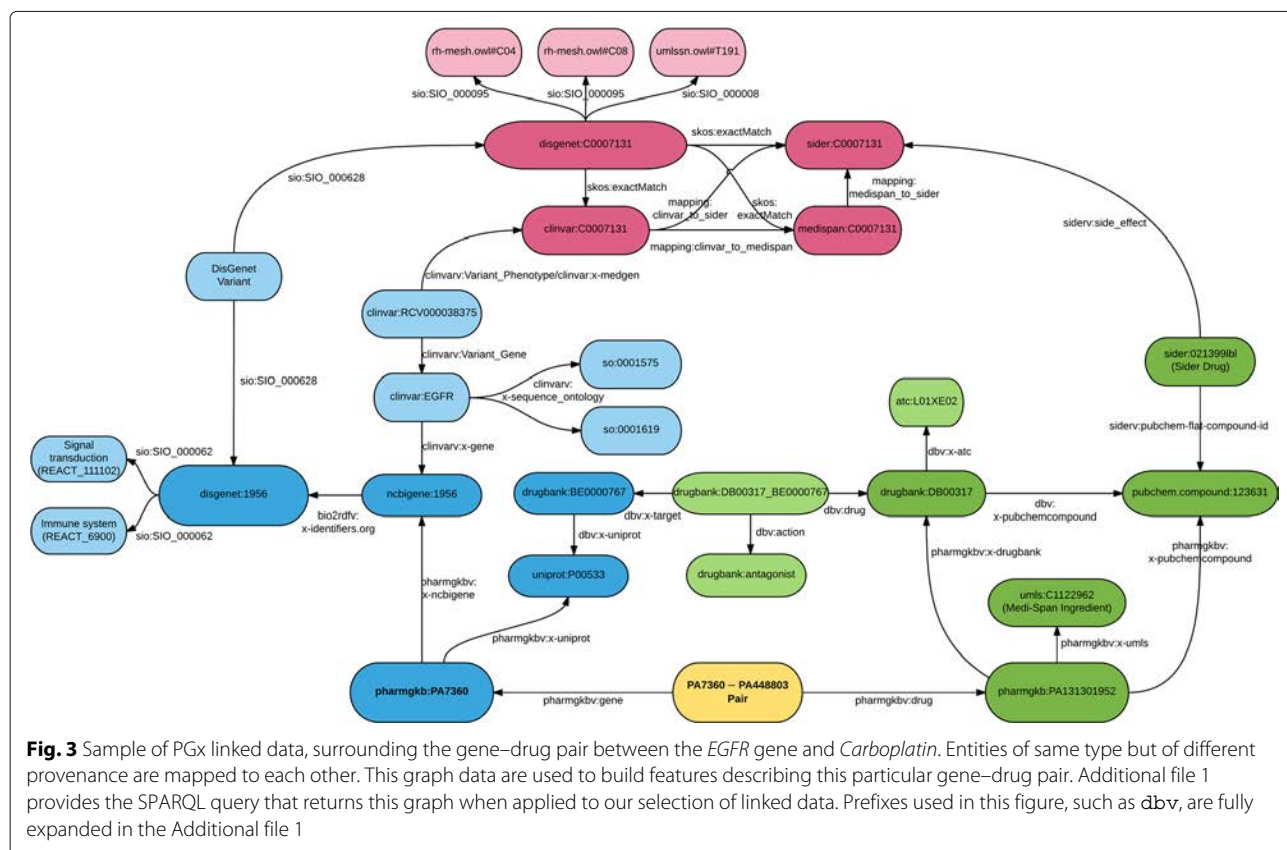


**Fig. 1** Overview of the type of entities and relationships considered and their origin. Entities are of three distinct types: Gene, Phenotype and Drug. Gene–Phenotype relationships are coming from ClinVar and DisGeNET, Phenotype–Drug relationships from SIDER and Medi-Span, Gene–Drug relationships from DrugBank. In addition, we included gene and drug entities from PharmGKB to enable building the training and test sets. Equivalence mappings are defined between entities of the same type but of different origin. In addition to entity–entity relationships, we consider some attributes that are specific to entities, such as the ATC class of drug that is a drug attribute. Naming of different parts of the data (e.g., G–P links, gene attributes) is used later in the step of formatting of the linked data. The detailed schema of the data is provided Fig. 2

**Data RDFization** The second step is about turning selected data in a standardized RDF graph, available at <https://pgxlod.loria.fr>. We benefit from the fact that DisGeNET [44], SIDER [45] and DrugBank [46] are already available online in the form of LOD and reused them. DisGeNET includes data from ClinVar, but because it includes only a part of it, we made our own RDF version of ClinVar following guidelines and scripts of the Bio2RDF project. We completed the Bio2RDF version of PharmGKB locally with gene–drug relationships manually annotated by PharmGKB but not openly distributed [15]. Similarly, we transformed drug indications and side-effects from Medi-Span in the form of RDF triples and loaded them into our SPARQL server. For

**Mapping definition** To define mappings, we first relied on standard identifiers such as NCBI Gene ID found in DisGeNET and ClinVar URIs and UMLS CUI found in DisGeNET, ClinVar, SIDER and Medi-Span. We defined regular expressions over URIs to isolate identifiers and when two match, we define a mapping. Figure 3 shows two





entities, *clinvar:1956* and *disgenet:1956* that share a unique identifier within different namespaces.

Second, when no standard identifier exists, we relied on services provided by *biodb.jp* to obtain cross-references between identifiers and, accordingly, define mappings [47]. We implemented a tool named *biojp2rdf* that transforms the cross-references provided by *biodb.jp* in RDF [48]. For drugs, we also relied on the API provided by *RxNav* to obtain mappings between *Medi-Span* identifiers and *UMLS CUIs* [49]. We loaded all mapping data into our SPARQL server to enable the resolution of identity between entities of the same type.

### Learning task, training and test sets

**Learning task** Our learning task is a supervised classification of specific nodes from the data graph. Nodes considered for classification represent candidate pairs of one gene and one drug, which are binary classified as either pharmacogenomically associated or not. Each candidate pair is added to the data graph as a new node linked to both the gene and the drug constituting the pair. Figure 3 provides an example of such a pair and its links to its constituents.

**Training set** To constitute our training set we defined two sets of instances: *positives* and *negatives*. Our set

of positives is gene–drug pairs annotated as associated according to *PharmGKB* (version of October 1<sup>st</sup>, 2015) and that are annotated with a high level of validation in *PharmGKB*, i.e., level= 1 or 2 [14]. *PharmGKB* clinical annotations are relating gene variants to drugs, not gene to drugs. We generalized these relationships to manipulate gene–drug relationships. When 2 variants of a same gene are associated with two distinct levels of evidence to the same drug, we conserve only the highest. For instance, the *VKORC1* gene has several variants associated to *warfarin* with level of evidence from 1 to 4. We conserve only that the *VKORC1* gene is associated to *warfarin* with a level of evidence 1. Accordingly, we generated 91 positive instances.

To constitute our set of negatives, we randomly generated gene–drug pairs from those listed in *PharmGKB*, but checked to be absent from *DGIdb* (the Drug Gene Interaction database), which collects gene–drug relationships from various sources [21], including *PharmGKB*. Two distinct sets of negatives were generated, one of 91 instances to balance exactly with the number of positives and one of 182, to experiment with this unbalance.

**Test set** We considered the 1760 gene–drug pairs insufficiently validated according to *PharmGKB*, i.e., associated with a level of evidence 3 or 4.



### Learning pharmacogenes with Random Forests

**The Random Forest algorithm** Introduced by Leo Breiman in 2001 [50], Random Forest (RF) is an ensemble method, combining decision trees in order to obtain better results in supervised learning tasks. Let  $X$  and  $Y$  compose a training set, where  $X = \langle x_1, x_2, \dots, x_n \rangle$  is a vector of feature vectors and  $Y$  is a vector of classes  $Y = \langle y_1, y_2, \dots, y_n \rangle$ . A class  $y_i$  is accordingly associated with each feature vector  $x_i$ . In the case of a binary classification, each vector  $x_i$  is associated with a value  $y_i$  that is either 0 or 1. The method begins by creating several new learning sets, each one being a sample –with replacement– of elements from  $X$ ; described by their classes and a sample of their features. A decision tree is then trained on each learning set to take part in a majority vote, which result is the result of the RF. This approach enables a better accuracy and generalization of the model, and counterbalances the instability of decision trees, a forest being more stable to slight changes in the data.

**Data formatting with RF** Linked data are in the form of graphs, whereas machine learning algorithms such as RF take a feature matrix as an input. Consequently, our PGx linked data requires to be formatted in the form of such a matrix. Each line of a feature matrix represents an instance and each column represents a feature describing the instances. We propose encoding parts of the RDF graph by observable paths that start from the gene and the drug of a gene–drug pair. These paths start from the gene or the drug and potentially reach each others. To contain the size of the matrix, we simplify paths from genes and drugs in path of length 1, hereafter named *G–D link*, *G–P link* or *D–P link*, depending on the entity that are connected. In addition to these links, we encode few *attributes* that qualify drugs, genes themselves and phenotypes that are connected to them through G–P or D–P links. Figure 1 summarizes the elements of the graph we consider in this formatting step. Because several paths may leave a gene or drug, one gene–drug pair may be described in the matrix by several instances. But each instance describes a unique pair. A pair is thus described by the set of instances that represent possible combinations of paths and attributes associated with a pair. As

an example, Table 1 shows the matrix obtained when formatting the sample of linked data represented in Fig. 3.

### Multi-instance classification and candidate ranking

With RF classification and GK, a probability distribution value, denoted  $p_{RF}$  or  $p_{GK}$ , may be used to evaluate the confidence of the model for classifying a new instance and then rank classified instances. However, the gene–drug pairs that we classify are typical examples of multi-instance objects, also named bag of instances, since they are not represented by a single instance but by several ones. Additional treatment is then required to classify and rank bags of instances. One option, as seen in [51], is to use the max operator, such that  $p_i = \max p_{ij}$ , where  $p_i$  is the probability estimate for the bag  $i$ , and  $p_{ij}$  the probability estimates of all instances  $j$  of the bag  $i$ . Another option would be to compute the arithmetic mean  $\bar{p}$  of probabilities of instances of the bag [52]. However, one bag of instances can contain at the same time instances classified as positive and instances classified as negative in our case, all with a high  $p_{ij}$ . In this case, applying the max or the arithmetic mean operator would lead to false positives. We choose to use a weighted mean to aggregate all the  $p_{ij}$ . Let  $n_i$  be the number of instances in the bag  $B_i$  and  $Class_{ij}$  the classification decision proposed by the model for the instance  $j$  of the bag  $B_i$ .

$$p_i = \frac{\sum_{j=1}^{n_i} a \times p_{ij}}{n_i}, \text{ with } a = \begin{cases} -1 & \text{if } Class_{ij} = 0 \\ 1 & \text{if } Class_{ij} = 1 \end{cases} \quad (1)$$

For each bag  $B_i$ ,  $p_i \in [-1, 1]$ . If every instance of a bag is associated with a strong confidence for being classified as positive ( $Class_{ij} = 1$ ), then  $p_i$  will be close to 1. In the case of a bag of instances associated with a strong confidence for a negative classification,  $p_i$  will be close to -1.  $p_i$  close to 0 means that we cannot classify, positively or negatively, the bag with a strong confidence.

### Learning pharmacogenes with graph kernels

**Data formatting for graph kernel** Graph Kernels (GK) present the advantage of handling directly data in the form of graphs. In addition, the Mustard library that we

**Table 1** Example of a feature matrix generated from linked data

ID	Gene attribute	Phenotype	Drug attribute	G-D link	G-P link	D-P link	Class
PA7360-PA131301952	Signal transduction	C0007131	L01XE2	Antagonist	clinvar: so_0001575	sider: indication	1
PA7360-PA131301952	Immune system	C0007131	L01XE2	Antagonist	clinvar: so_0001575	sider: indication	1
PA7360-PA131301952	Signal transduction	C0007131	L01XE2	Antagonist	clinvar: so_0001619	sider: indication	1
PA7360-PA131301952	Immune system	C0007131	L01XE2	Antagonist	clinvar: so_0001519	sider: indication	1

All the instances (e.g., lines) describe the same gene–drug relationships (EGFR–Gefitinib), which is associated in PharmGKB with a high level of evidence ( $Class=1$ ). Figure 3 shows some of the data associated with this relationships in linked data. Values are extracted from the graph and are encoded in various manner. For example, values of phenotypes are UMLS CUI, values of drug attributes are ATC codes

propose using, handles directly RDF graphs as an input, consequently limiting formatting efforts. GK generates the attributes of the instances either as a list of feature vectors or as a kernel matrix. Consequently, we provide to Mustard the PGx linked data that we selected. Mustard is designed to compute RDF node classification, whereas we want computing link prediction. To adapt to Mustard, we enriched the PGx data with artificial entities so we can adapt link prediction to a classification task. Concretely, we add entities to represent gene–drug relationships, each related to a unique gene and drug. In addition, two classes named *associated* and *not associated* are added to the graph and related with pairs of the training set. For example, a positive pair from the training set will be a node in RDF, related to the class named *associated*. Mustard task is to classify test pairs as associated or not. Finally, we needed to invert the direction of some links in our graph since Mustard allows exploring predicates in only one direction, whereas we want kernel functions exploring the full graph without considering the direction of predicates. Enriching the graph with inverse predicates has been considered but this generates many cycles that are indeed considered by some of the graph substructures then computed by the kernel functions (see the next subsection for details).

### Graph kernels in Mustard

In [34], de Vries et de Rooij proposed a general framework, implemented within the Mustard library, with a list of kernels to generate the features of RDF instances. The framework works as follows: First, the neighborhood of each instance, up to a certain depth, is extracted. Additional file 2 proposes an example of SPARQL query that returns the neighborhood, up to a length of 4, of an example drug. Then, predefined substructures are counted within the boundaries of this neighborhood. The attributes of each instance are then the count, for that instance, of the substructures extracted from all neighborhoods.

Mustard includes 3 different types of substructures defined below (see [34] for more details):

- *Bag of labels*: A bag of labels is simply the set of vertex labels in the instance neighborhood.
- *Walks*, up to a certain length: A walk is a set of consecutive edges in the graph.
- *Sub-trees*, up to a certain length: A sub-tree originating at a vertex is the acyclic graph around that vertex.

Note that the description of the substructures does not require the distinction made in RDF graphs between labels of nodes and labels of edges. Indeed the substructure counting algorithms take care of reifying the

edges and transforming them to labeled nodes. By varying the size of the neighborhood and precisising the type of substructures, one obtain various feature vectors. We list below parameters that can be changed to compute kernels:

- *Exploration depth*: This parameter defines the depth of instance neighborhood from which we extract and count the substructures.
- *Cycles traversal*: During exploration, cycles can be traversed either once, or multiple times. In the latter case, the same nodes in the cycle get repeated, and the obtained neighborhood is a *Tree* (an acyclic graph) rooted at the instance vertex. Otherwise the neighborhood is considered as a *Sub-graph*.
- *Root constraint*: If the neighborhood is a tree, we also consider the constraint in which only substructures that start from the root vertex are counted. This can lead to a faster computation.
- *Substructure depth*: When counting substructures, we can define the maximum length (respectively depth) of the walks (resp. sub-trees).
- *Minimum frequency*: Two of the main differences between RDF graphs and theoretical graphs usually considered in graph mining are that vertices and nodes in RDF have labels, and there is a large number of different labels in the graph. Many labels may be used only once. This leads, if considering labels, to very specific graph patterns, which do not generalize well. To alleviate this problem, Mustard enables imposing a minimum frequency, under which a label is not counted.

The obtained feature vectors are very sparse, which are efficiently computed using matrix dot products, called kernels. Those kernels are adapted to be computed by Support Vector Machine (SVM) algorithm that are classically the learning algorithm on the basis of graph kernels. We used this algorithm in this work.

## Results and interpretation

### Random Forest results

We trained and evaluated our model using the Weka implementation of the RF and a 10-fold cross-validation. First, we performed an information gain analysis, classically used for feature selection, and found out that the feature named *disease attribute* was providing little information to the classifier (*InfoGain* = 0.008). We decided to remove this feature from both the training and test sets. Table 3 presents the results of the evaluation on the model trained with unbalanced data, and Table 2 the evaluation on the model trained with balanced data.

We evaluate two concurrent models trained either with a balanced set of positive and negative pairs (resp. 91

**Table 2** Results of the 10-fold cross-validation of our first RF model

Class	Precision	Recall	F-Measure
1 (positive)	0.944	0.904	0.924
0 (negative)	0.997	0.998	0.998
Weighted Average	0.996	0.996	0.996

This model is trained with 91 positive and 91 negative gene–drug relationships

and 91) or an unbalanced set (resp 91 and 182). The model with balanced classes clearly overfits, with a F-measure=0.996 (see Table 2). This is most probably due to the fact that when formatting the data, negative pairs generate much less instances, leading to a large unbalance between positive (108,038) and negative (3197) instances in the feature matrix. A larger set of negative pairs, leading to a more balanced feature matrix, may temper the overfitting. In this case we obtained a F-measure of 0.729 (see Table 3) and a root mean squared error of 0.385.

With this last model, we classified the 1760 pairs (represented by 984,460 instances) of our test set. The top-20 pairs predicated as positive according to our RF model are provided online at [53, 54].

### Graph kernel results

We used the Mustard library as an implementation of a GK framework to perform the two next experiments.

**Evaluating the impact of GK parameters** The purpose of the first experiment is to evaluate the impact of various kernel settings on the RDF graph we consider here. For each kernel a C-SVC (C-Support Vector Classification) support vector machine from the LibSVM library is trained. Each kernel is evaluated with a 10-fold cross-validation, which is repeated 10 times itself with different randomization seeds. Within each fold, SVM parameters are optimized, again using a 10-fold cross-validation.

Table 4 illustrates how the F-measure of our model changes depending on the substructure and the type of neighborhood considered. Table 5 compares F-measures obtained with various neighborhood depth, for a fixed substructure and type of neighborhood. Surprisingly, the F-measure is not strongly impacted by this parameter.

**Table 3** Results of the 10-fold cross-validation of our second RF model

Class	Precision	Recall	F-Measure
1 (positive)	0.804	0.998	0.891
0 (negative)	0.994	0.547	0.706
Weighted Average	0.728	0.735	<b>0.729</b>

This model is trained on 91 positive and 182 negative gene–drug relationships

**Table 4** Comparison of F-measures obtained with various combination of substructure and neighborhood settings.

F-measures are averaged over two other parameters: the *depth* of the neighborhood ( $d = 4, 6, 8, 10, 12, 15$ ) and the *length* of the substructures ( $l = 4, 6, 8, 10, 12, 15$ )

Substructures \ Neighborhood	Graph	Tree
Bag of Labels	0.763	0.781
Walks	0.777	0.797
Sub-trees	0.782	<b>0.803</b>

We think that this is due to the fact that most important features are in a distance of 4. Table 6 illustrates how F-measures can be impacted by the root constraint. Table 7 shows the impact on the F-measure of imposing a minimum frequency to the type of vertices and edges considered in the mining.

**Classifying candidate pharmacogenes** In the second experiment, we trained our model and applied it to our test set to classify candidate pharmacogenes. Regarding the evaluation of the model, a 10-fold cross-validation is done and repeated 10 times with different randomization seeds. Within each fold, we optimize the different kernel settings again using 10-fold cross-validation. The reason for optimizing the kernel settings within an inner cross-validation instead of the selecting the best settings from the previous experiment, is to avoid a model selection bias which can lead to a misleading optimistic performance evaluation as shown in [53, 54].

Table 8 presents the results of the evaluation of the model trained with both balanced and unbalanced data. We report the F-measure for the positive class, the average F-measure and the AUC-ROC. For the best average F-measure, i.e., 0.807, the error rate for a 95% confidence interval is 0.008.

With the unbalanced model, we classified the 1760 instances of our test set. The top-20 pairs predicated as positive according to our GK model are provided online at [55].

### Result combination

We intersected the two lists of top candidate pairs obtained by RF and GK, to keep only those present in both classification. Then, we sorted the pairs by descending order of  $p_{RF}$ . Table 9 presents the 20-top candidates obtained by this method. We notice that with this ranking

**Table 5** Comparison of F-measures obtained with different depths of neighborhood

$d=4$	$d=6$	$d=8$	$d=10$	$d=12$	$d=15$
0.807	0.805	0.801	0.803	0.803	0.804

Neighborhood setting=Tree and substructures=Sub-trees



**Table 6** Comparison of F-measures obtained with and without the *root constraint*

	w/ Root constraint	w/o Root constraint
Bag of Labels	0.479	0.781
Walks	0.753	0.797
Sub-trees	0.536	<b>0.803</b>

F-measures are averaged over the *length* of the substructures ( $l = 4, 6, 8, 10, 12, 15$ )

both  $p_{RF}$  and  $p_{GK}$  are closed to 1. A PGx expert (NCN) examined the 20-top candidates within a manual literature study to evaluate their relevance and estimate their interest for further investigation. Results of this examination are reported in the next subsection.

### Interpretation

Among the top-20 candidates obtained with both predictions models (Table 9), unreleased gene–drug pairs which should be further investigated were combined to extensively-studied candidates, not surprisingly mainly in cancerology.

For instance, it is widely known that aberrant epidermal growth factor receptor (EGFR) signaling lead to various oncogenic phenotypes [56] and previous PGx investigations have shown that the *EGFR* gene mutation status was associated with EGFR-targeted agents efficacy such as Erlotinib's (rank 2) in the case of non-small cell lung cancer (NSCLC) [56, 57]. In addition to its single agent activity, it has also been shown that this tyrosine kinase inhibitor acts in synergy with standard chemotherapy such as Fluorouracil in various cancer patients [58, 59] and we were able to pair Fluorouracil with the *EGFR* gene as well (rank 3).

Conversely, the *MAP3K1* gene's association with Carboplatin (rank 1) seemed novel and yet, in a genome-wide association study on advanced NSCLC patients treated with this antineoplastic chemotherapy drug, a single nucleotide polymorphism in the *DSCAM* gene has been identified as a prognostic biomarker candidate [60]. This supports our drug-gene pair in rank 6 and gives insights on possible *MAP3K1*  $\times$  *DSCAM* synergy that should be further investigated.

Those various outputs (confirming bibliography or unreleased) show that our approach could be of value in 1) strengthening PGx knowledge and facilitate its translation in practice and 2) leading to novel investigations in order to better identify the complex synergies in action.

**Table 7** Comparison of F-measures obtained with different *minimum frequency* settings

Min. frequency	2	4	8	16	32
F-measure	0.794	0.780	0.798	0.796	0.774

Neighborhood setting=Tree and substructures=Walks

**Table 8** Results of the 10-fold cross-validation of our Graph Kernel/SVM model

	F-measure	Avg. F-measure	AUC-ROC
Balanced	0.770	0.761	0.840
Unbalanced	0.746	<b>0.807</b>	0.905

Models are trained with 91 positive and 91 negative examples for the *Balanced* model and 91 and 182 for the *Unbalanced*

### Discussion

We considered first the RF algorithm because it has been successively applied for the prediction of drug–drug interactions from a set of RDF statements [37, 48]. The availability of the Mustard library and its results in term of node classification motivates us to compare RF with GK. One drawback of the Graph Kernel method is that it is not always possible to know which part of the graph data have the biggest contribution to classification since a graphs is classified by similarity. This may motivate the investigation of other subgraph mining methods that may be more informative on the weights of substructures in the classification. One may consider techniques such as gBoost [61] that progressively collects informative patterns, or gSpan [62] that enumerates frequent subgraph used as features for classification.

**Table 9** 20-Top candidates of gene–drug pairs predicted from our PGx linked data

Rank	Gene	Drug	$p_{RF}$	$p_{GK}$
1	<i>MAP3K1</i>	Carboplatin	0.993	0.991
2	<i>EGFR</i>	Erlotinib	0.992	0.980
3	<i>EGFR</i>	Fluorouracil	0.989	0.966
4	<i>FCER1G</i>	Aspirin	0.988	0.993
5	<i>MAP3K1</i>	Erlotinib	0.988	0.830
6	<i>DSCAM</i>	Carboplatin	0.979	0.974
7	<i>CHIA</i>	Aspirin	0.979	0.911
8	<i>GP6</i>	Aspirin	0.979	0.976
9	<i>ACE</i>	Sildenafil	0.979	0.911
10	<i>TPMT</i>	Cyclophosphamide	0.975	0.994
11	<i>CYP2B6</i>	Nicotine	0.973	0.912
12	<i>PTGER3</i>	Aspirin	0.967	0.965
13	<i>NTRK1</i>	Aspirin	0.967	0.992
14	<i>EXO1</i>	Fluorouracil	0.966	0.694
15	<i>ERBB2</i>	Trastuzumab	0.964	0.998
16	<i>CYP2B6</i>	Olanzapine	0.964	0.965
17	<i>HLA-DQ1</i>	Azathioprine	0.963	0.931
18	<i>HMGR</i>	Simvastatin	0.963	0.996
19	<i>CYBA</i>	Simvastatin	0.961	0.973
20	<i>HLA-DRB1</i>	Mercaptopurine	0.961	0.966

RF algorithm performs correctly ( $F\text{-m} = 0.73$ ) in the frame of our case study, but presents several limitations. First, RF is limited by our usage of a multi-instance representation of data, i.e., data about one gene–drug pair is represented in the feature matrix by several lines [52]. Because our dataset contains much more data about positive examples than data about negatives, it results that if we balance the number of positive and negative examples in the training set (respectively 91 and 91), the actual number of lines (i.e., instances) in the matrix describing positive examples is much larger than for negatives (respectively 108,038 and 3,197). However, our experiments showed that initial selection of negative examples, as well as keeping a certain unbalance, is important. This large unbalance lead the RF to an overfitting, i.e., an instance to classify will most probably be similar to one of the numerous descriptions of positive examples and be classified as positive. We overcome this drawback by doubling the number of negative examples in our training set. This unbalancing of examples (respectively 91 positives and 182 negatives) resulted in a reduction of the unbalance in the feature matrix (respectively 108,038 and 57,885). Here we can note that the second set of negative example is described by much more instances than the first one, what let us see that our random pick of negative examples in the large set of gene–drug relationships not referenced by DGIdb impacts to some extent the results of our approach.

Another limitation of RF is that it requires a formatting of the graph data and then to select a set of features. We achieved this selection manually to retain 6 features, and then apply a standard feature selection method (Information Gain) that enabled us to filter one useless feature (disease attribute such as the MeSH class of the disease).

Our experiment shows that GK achieves globally better than RF, and that the Mustard library offers many facilities to mine RDF data. In addition, it provides us several insights on the features that are more relevant to consider when mining our PGx linked data. For example, it seems that considering only the close neighborhood ( $d=4$ ) of instances to classify is sufficient in our case. Also, in the case of a constrained graph, as the one we designed, considering substructure in the neighborhood of instance may not be of primary importance. This may be associated to the fact that we limited the size and connectivity of the data graph, and consequently knowing solely the label of a set of edges and of vertices may enable to reconstruct a path or a subgraph in the neighborhood of an instance.

Our selection of data sources may be discussed, particularly because some of those are not open. Of course adding new sources would be of interest. Indeed, our choice for the linked data framework is motivated by the fact that we want to ease the addition/removal of data

sources for enabling the selection of best features out of many sources without considering if they are open or not. In regards with the results from previous works [40, 41], we think that sources of triples extracted from the literature would be particularly valuable, such as those extracted in [25]. Because GK considers data directly in the form of a graph, one could want to mine directly LOD resources, without particular selection. However, the large number of available data in the LOD, including many non-informative metadata, makes this still challenging. We decided to use multiple data sources, with various license agreement. Further work could evaluate the impact of adding/removing data sources, then offering the opportunity to compare the importance of open data vs. not open data.

A limitation to our approach is related to the field of PGx itself since only few (91) gene–drug relationships have a high level of evidence according to PharmGKB, making our training set relatively small. One way of enlarging the size of the training set would be to consider *gene variant–drug* relationships, instead of gene–drug, that have two advantages: being more numerous, but also rendering more precisely the state of the art of PGx knowledge. Indeed, a gene may host two (or more) variants, one that impacts drug response and one that does not.

Two biases are to consider when interpreting the results. First, negative examples are gene–drug relationships not listed in DGIdb, which includes known and predicted candidates from many databases. Consequently, negatives are likely not to be related, instead of not being related, but to our knowledge no existing resource lists negative gene–drug relationships. Second, tested examples are likely to be related since they are listed in PharmGKB with a low level of confidence. However, our goal is prioritize these candidates, instead of detecting negatives from those.

## Conclusion

This article is a proposal to help validating candidate pharmacogenes by learning from PGx linked data. More precisely, we selected and interconnected data relevant to the PGx domain in the form of a large RDF graph. Then, we formatted these data to train and compare a RF and a GK classifier. These two classifiers were evaluated and used to identify and rank candidate pharmacogenes. GK achieves a F-measure of 0.81, whereas RF reaches 0.73. Top candidate pharmacogenes pointed out by our approach are provided and interpreted in this article. Top candidates that are not already extensively studied will be further investigated by PGx experts. Results we obtained with the GK library named Mustard are particularly promising both for our application domain, i.e., validating pharmacogenes, and more broadly for the mining of biomedical linked data.

## Additional files

**Additional file 1:** SPARQL query example 1. This text file contains the SPARQL query we apply on our PGx linked data to obtain the data graph represented in Fig. 3. This query includes the definition of prefixes mentioned in Figs. 2 and 3. This query takes about 30 s on our <https://pgxlod.loria.fr> server. (TXT 2 kb)

**Additional file 2:** SPARQL query example 2. This text file contains an example of SPARQL query that enable to explore the vicinity of an entity. This particular query returns the RDF graph surrounding, within a length of 4, the node `pharmgkb:PA451906` that represents the *warfarin*, an anticoagulant drug. (TXT 392 bytes)

## Abbreviations

AUC-ROC: Area under the receiver operating characteristic curve; GK: Graph kernel; LOD: Linked open data; PGx: Pharmacogenomics; RDF: Resource description framework; RF: Random forest; SPARQL: SPARQL protocol and RDF query language; SVM: Support vector machine; URI: Uniform resource identifier

## Acknowledgements

We acknowledge the participants of the SWAT4LS 2015 conference for their constructive feedback on the preliminary results of this work.

## Funding

This project is supported by the *PractikPharma* project, grant ANR-15-CE23-0028, funded by the French National Research Agency (<http://practikpharma.loria.fr/>) and by *Snowflake*, an Inria associate team (<http://snowflake.loria.fr/>).

## Availability of data and materials

Elements of software developed for this work are available at [48] and [63]. Our set of PGx linked data is available at <https://pgxlod.loria.fr/>. An account will be provided upon request.

## Authors' contributions

KD, YM and AC designed the experiments and wrote the manuscript. KD and YM developed necessary code and ran the experiments. SDS and NCN edited the manuscript. SDS advised in the manipulation and interpretation of RF and GK. NCN advises about the pharmacogenomic use case and interprets the results. PR set up the data server and advised about technical aspects of the work. All authors read and approved the final manuscript.

## Competing interests

The authors declare that they have no competing interests.

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Author details

<sup>1</sup>LORIA (CNRS, Inria Nancy-Grand Est, University of Lorraine), Campus Scientifique, Nancy, France. <sup>2</sup>Ecole nationale supérieure des mines de Nancy, Campus Artem, Nancy, France. <sup>3</sup>UMR U1122 IGE-PCV (INSERM, University of Lorraine), 30 Rue Lionnois, Nancy, France.

Received: 9 June 2016 Accepted: 29 March 2017

Published online: 20 April 2017

## References

- Xie HG, Frueh FW. Pharmacogenomics steps toward personalized medicine. *Personalized Med.* 2005;2(4):325–7.
- Garten Y, Coulet A, Altman RB. Recent progress in automatically extracting information from the pharmacogenomic literature. *Pharmacogenomics.* 2010;11(10):1467–89.
- Whirl-Carrillo M, et al. Pharmacogenomics knowledge for personalized medicine. *Clin Pharmacol Ther.* 2012;92(4):414–17.
- Ioannidis JPA. To replicate or not to replicate: The case of pharmacogenetic studies. *Circ Cardiovasc Genet.* 2013;6:413–8.
- Zineh I, Pacanowski M, Woodcock J. Pharmacogenetics and coumarin dosing? Recalibrating expectations. *N Engl J Med.* 2013;369:2273–5.
- Bizer C, Heath T, Berners-Lee T. Linked data - the story so far. *Int J Semantic Web Inf Syst.* 2009;5(3):1–22.
- Antezana E, Kuiper M, Mironov V. Biological knowledge management: the emerging role of the Semantic Web technologies. *Brief Bioinform.* 2009;10(4):392–407.
- Callahan A, Cruz-Toledo J, Ansell P, Dumontier M. Bio2rdf release 2: Improved coverage, interoperability and provenance of life science linked data. In: *Proceedings of the 10th European Semantic Web Conference, ESWC 2013. Lecture Notes in Computer Science 7882.* Springer; 2013. p. 200–12.
- Jupp S, Malone J, Bolleman J, Brandizi M, Davies M, Garcia LJ, Gaulton A, Gehant S, Laibe C, Redaschi N, Wimalaratne SM, Martin MJ, Novère NL, Parkinson HE, Birney E, Jenkinson AM. The EBI RDF platform: linked open data for the life sciences. *Bioinformatics.* 2014;30(9):1338–9.
- Kinjo AR, et al. Protein Data Bank Japan (PDBJ): maintaining a structural data archive and resource description framework format. *Nucleic Acids Res.* 2012;40(Database issue):D453–60.
- Samwald M, Jentzsch A, Bouton C, Kallesøe CS, Willighagen E, Hajagos J, Marshall MS, Prud'hommeaux E, Hassenzadeh O, Pichler E, Stephens S. Linked open drug data for pharmaceutical research and development. *J Cheminformatics.* 2011;3:19. Accessed 07 June 2016.
- Good BM, Wilkinson MD. The life sciences semantic web is full of creeps! *Brief Bioinform.* 2006;7(3):275–86.
- Marshall MS, Boyce R, Deus HF, Zhao J, Willighagen EL, Samwald M, Pichler E, Hajagos J, Prud'hommeaux E, Stephens S. Emerging practices for mapping and linking life sciences data using RDF — A case series. *Web Semant Sci Serv Agents World Wide Web.* 2012;14:2–13. Accessed 07 June 2016.
- PharmGKB. Levels of evidence of annotations. <https://www.pharmgkb.org/page/clinAnnLevels>. Accessed 1 June 2016.
- Bio2RDF project. PharmGKB endpoint. <http://cu.pharmgkb.bio2rdf.org/sparql>. Accessed 1 June 2016.
- Wishart DS, Knox C, Guo A, Cheng D, Shrivastava S, Tzur D, Gautam B, Hassanali M. Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Res.* 2008;36(Database-Issue):901–6.
- Landrum MJ, Lee JM, Riley GR, Jang W, Rubinstein WS, Church DM, Maglott DR. Clinvar: public archive of relationships among sequence variation and human phenotype. *Nucleic Acids Res.* 2014;42(Database-Issue):980–5.
- Kuhn M, Campillos M, Letunic I, Jensen LJ, Bork P. A side effect resource to capture phenotypic effects of drugs. *Mol Syst Biol.* 2010;6(1):343.
- Kuhn M, Letunic I, Jensen LJ, Bork P. The sider database of drugs and side effects. *Nucleic Acids Res.* 2016;44(D1):D1075–9.
- Kuhn M, Letunic I, Jensen LJ, Bork P. The SIDER database of drugs and side effects. *Nucleic Acids Res.* 2016;44(Database-Issue):1075–9. doi:10.1093/nar/gkv1075.
- Wagner AH, Coffman AC, Ainscough BJ, Spies NC, Skidmore ZL, Campbell KM, Krysiak K, Pan D, McMichael JF, Eldred JM, Walker JR, Wilson RK, Mardis ER, Griffith M, Griffith OL. Dgidb 2.0: mining clinically relevant drug-gene interactions. *Nucleic Acids Res.* 2016;44(Database-Issue):1036–44.
- Piñero J, Queralt-Rosinach N, Bravo A, Deu-Pons J, Bauer-Mehren A, Baron M, Sanz F, Furlong LI. Disgenet: a discovery platform for the dynamical exploration of human diseases and their genes. *Database.* 2015;2015:bav028.
- Samwald M, Coulet A, Huerfano I, Powers RL, Luciano JS, Freimuth RR, Whipple F, Pichler E, Prud'hommeaux E, Dumontier M, Marshall MS. Semantically enabling pharmacogenomic data for the realization of personalized medicine. *Pharmacogenomics.* 2012;13(2):201–12.
- Hoehndorf R, Dumontier M, Gkoutos GV. Identifying aberrant pathways through integrated analysis of knowledge in pharmacogenomics. *Bioinformatics.* 2012;28(16):2169–75.
- Coulet A, Garten Y, Dumontier M, Altman RB, Musen MA, Shah NH. Integration and publication of heterogeneous text-mined relationships on the semantic web. *J Biomed Semant.* 2011;2(S-2):10.
- Bicer V, Tran T, Gossen A. Relational kernel machines for learning from graph-structured RDF data. In: *Proceedings of the 8th Extended Semantic Web Conference, Part I, ESWC 2011. Lecture Notes in Computer Science 6643.* Springer; 2011. p. 47–62.
- Huang Y, Trespo V, Bundschuh M, Rettinger A, Kriegel H. Multivariate prediction for learning on the semantic web. In: *Proceedings of the 20th*

- International Conference on Inductive Logic Programming, ILP 2010. Lecture Notes in Computer Science 7489. Springer; 2010. p. 92–104.
28. Thor A, Anderson P, Raschid L, Navlakha S, Saha B, Khuller S, Zhang XN. Link Prediction for Annotation Graphs Using Graph Summarization. In: Proceedings of the 10th International Conference on The Semantic Web - Volume Part I ISWC'11. Springer; 2011. p. 714–29.
  29. Lösch U, Bloehdorn S, Rettinger A. Graph kernels for RDF data. In: Proceedings of the 9th Extended Semantic Web Conference, ESWC 2012. Lecture Notes in Computer Science 7295. Springer; 2012. p. 134–48.
  30. de Vries GKD. A fast approximation of the weisfeiler-lehman graph kernel for RDF data. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, Part I, ECML PKDD 2013. Lecture Notes in Computer Science 8188. Springer; 2013. p. 606–21.
  31. Brenninkmeijer CYA, Dunlop I, Goble CA, Gray AJG, Pettifer S, Stevens R. Computing identity co-reference across drug discovery datasets. In: Proceedings of the 6th International Workshop on Semantic Web Applications and Tools for Life Sciences, SWAT4LS 2013. CEUR Workshop Proceedings 114.
  32. Volz J, Bizer C, Gaedke M, Kobilarov G. Discovering and maintaining links on the web of data. In: Proceedings of the 8th International Semantic Web Conference, ISWC 2009. Lecture Notes in Computer Science 5823. Springer; 2009. p. 650–65.
  33. Heim P, Lohmann S, Stegemann T. Interactive relationship discovery via the semantic web. In: Proceedings of the 7th Extended Semantic Web Conference, Part I, ESWC 2010. Lecture Notes in Computer Science 6088. Springer; 2011. p. 303–17.
  34. de Vries GKD, de Rooij S. Substructure counting graph kernels for machine learning from RDF data. *J Web Sem*. 2015;35:71–84.
  35. Kondor R, Lafferty JD. Diffusion kernels on graphs and other discrete input spaces. In: Proceedings of the 19th International Conference on Machine Learning, ICML 2002. Morgan Kaufmann; 2002. p. 315–22.
  36. Data2Semantics. Mustard – machine learning using svms to analyse rdf data, under mit licence. <https://github.com/Data2Semantics/mustard>. Accessed 01 June 2016.
  37. Percha B, Garten Y, Altman RB. Discovery and explanation of drug-drug interactions via text mining. In: PSB; 2012. p. 410–21. <http://psb.stanford.edu/psb-online/proceedings/psb2012/percha.pdf>.
  38. Dalleau K, Ndiaye NC, Coulet A. Suggesting valid pharmacogenes by mining linked data. In: Proceedings of the 8th Semantic Web Applications and Tools for Life Sciences International Conference, SWAT4LS 2015. CEUR Workshop Proceedings 1546; 2015. p. 49–58.
  39. Hansen N, Brunak S, Altman R. Generating genome-scale candidate gene lists for pharmacogenomics. *Clin Pharmacol Ther*. 2009;86(2): 183–9.
  40. Garten Y, Tatonetti NP, Altman RB. Improving the prediction of pharmacogenes using text-derived gene-drug relationships. In: PSB; 2010. p. 305–14. <http://psb.stanford.edu/psb-online/proceedings/psb10/garten.pdf>.
  41. Funk CS, Hunter LE, Cohen KB. Combining heterogeneous data for prediction of disease related and pharmacogenes. In: Pacific Symposium on Biocomputing; 2014. p. 328–39.
  42. Dumontier M, Villanueva-Rosales N. Towards pharmacogenomics knowledge discovery with the semantic web. *Brief Bioinform*. 2009;10(2): 153–63.
  43. Coulet A, Smail-Tabbone M, Napoli A, Devignes MD. Ontology-based knowledge discovery in pharmacogenomics. *Adv Exp Med Biol*. 2011;696: 357–66.
  44. DisGeNET endpoint. <http://rdf.disgenet.org/sparql/>. Accessed 01 June 2016.
  45. Bio2RDF project. SIDER endpoint. <http://cu.sider.bio2rdf.org/sparql>. Accessed 01 June 2016.
  46. Bio2RDF project. DrugBank endpoint. <http://cu.drugbank.bio2rdf.org/sparql>. Accessed 01 June 2016.
  47. Imanishi T, Nakaoka H. Hyperlink Management System and ID Converter System: enabling maintenance-free hyperlinks among major biological databases. *Nucleic Acids Res*. 2009;37(Web Server issue):17–22. Accessed 07 June 2016.
  48. Dalleau K. biojp2rdf – a tool to rdfize biodb.jp data, under mit licence. <https://github.com/KevinDalleau/biojp2rdf>. Accessed 01 June 2016.
  49. Zeng K, Bodenreider O, Kilbourne J, Nelson S. Rxnav: Towards an integrated view on drug information. In: Proceedings of the 12th World Congress on Health (Medical) Informatics, MEDINFO 2007. IOS Press 129; 2007. p. 386.
  50. Breiman L. Random Forests. *Mach Learn*. 2001;45(1):5–32. Accessed 2016-06-06.
  51. Leistner C, Saffari A, Bischof H. MIForests: Multiple-instance learning with randomized trees. In: Proceedings of the 11th European Conference on Computer Vision, Part IV, ECCV 2010. Lecture Notes in Computer Science 6316. Springer; 2010. p. 29–42.
  52. Amores J. Multiple instance classification: Review, taxonomy and comparative study. *Artif Intell*. 2013;201:81–105.
  53. 20-top candidate pharmacogenes, highlighted by our graph Random Forest classifier. [https://members.loria.fr/ACoulet/files/pgxlod/rf\\_20.csv](https://members.loria.fr/ACoulet/files/pgxlod/rf_20.csv). Accessed 11 Apr 2017.
  54. Cawley GC, Talbot NL. On over-fitting in model selection and subsequent selection bias in performance evaluation. *J Mach Learn Res*. 2010;11: 2079–107.
  55. 20-top candidate pharmacogenes, highlighted by our graph kernel / svm classifier. [https://members.loria.fr/ACoulet/files/pgxlod/gk\\_20.csv](https://members.loria.fr/ACoulet/files/pgxlod/gk_20.csv). Accessed 01 June 2016.
  56. Mayo C, Bertran-Alamillo J, Molina-Vila MA, Giménez-Capitán A, Costa C, Rosell R. Pharmacogenetics of EGFR in lung cancer: perspectives and clinical applications. *Pharmacogenomics*. 2012;13(7):789–802.
  57. de Mello RA, Madureira P, Carvalho LS, Araújo A, O'Brien M, Popat S. EGFR and KRAS mutations, and ALK fusions: current developments and personalized therapies for patients with advanced non-small-cell lung cancer. *Pharmacogenomics*. 2013;14(14):1765–77.
  58. Okabe T, Okamoto I, Tsukioka S, Uchida J, Iwasa T, Yoshida T, Hatashita E, Yamada Y, Satoh T, Tamura K, Fukuoka M, Nakagawa K. Synergistic antitumor effect of S-1 and the epidermal growth factor receptor inhibitor gefitinib in non-small cell lung cancer cell lines: role of gefitinib-induced down-regulation of thymidylate synthase. *Mol Cancer Ther*. 2008;7(3):599–606.
  59. Kim HK, Choi JJ, Kim CG, Kim HS, Oshima A, Yamada Y, Arai T, Nishio K, Michalowski A, Green JE. Three-gene predictor of clinical outcome for gastric cancer patients treated with chemotherapy. *Pharmacogenomics J*. 2012;12(2):119–27. Accessed 07 June 2016.
  60. Sato Y, Yamamoto N, Kunitoh H, Ohe Y, Minami H, Laird NM, Katori N, Saito Y, Ohnami S, Sakamoto H, Sawada JI, Saijo N, Yoshida T, Tamura T. Genome-wide association study on overall survival of advanced non-small cell lung cancer patients treated with carboplatin and paclitaxel. *J Thorac Oncol Off Publ Int Assoc Study Lung Cancer*. 2011;6(1):132–8.
  61. Saigo H, Nowozin S, Kadowaki T, Kudo T, Tsuda K. gboost: a mathematical programming approach to graph classification and regression. *Mach Learn*. 2009;75(1):69–89.
  62. Yan X, Han J. gspan: Graph-based substructure pattern mining. In: Proceedings of the 2002 IEEE International Conference on Data Mining, ICDM 2002 IEEE Computer Society; 2002. p. 721–4.
  63. Marzougui Y. pgx-lod-mining – adapting mustard to pgx linked data. <https://github.com/yassmarzou/pgx-lod-mining>. Accessed 28 Oct 2016.

Submit your next manuscript to BioMed Central  
and we will help you at every step:

- We accept pre-submission inquiries
- Our selector tool helps you to find the most relevant journal
- We provide round the clock customer support
- Convenient online submission
- Thorough peer review
- Inclusion in PubMed and all major indexing services
- Maximum visibility for your research

Submit your manuscript at  
[www.biomedcentral.com/submit](http://www.biomedcentral.com/submit)



## Quatrième partie

# Discussion et conclusion

Nous avons dans ce travail effectué une première sélection de sources de données d'intérêt en Pharmacogénomique, ouvertes et liées ou non, avant de toutes les transformer sous cette forme. Ces données, maintenant représentées sous le même formalisme et dans le même graphe, sont dorénavant accessibles *via* un point d'accès SPARQL dont l'accès est réservé : <https://pgxlod.loria.fr>. Nous avons par la suite appliqué deux méthodes d'apprentissage à ces données pour la prédiction de relations entre gènes et médicaments : les forêts d'arbres aléatoires et les machines à vecteur de support.

Plus précisément, la tâche d'apprentissage était ici la suivante : à partir de données sur des gènes, des phénotypes et des substances actives, classer une paire gène-médicament comme associée ou non d'un point de vue Pharmacogénomique. L'ensemble d'apprentissage est constitué de deux sous-ensembles d'Instances représentant des relations positives et des relations négatives. Notre ensemble de relations positives est constitué de paires gène-médicament annotées en tant que paires associées sur PharmGKB, et ayant un haut score de validation, *i.e.* de niveau 1 ou 2. Les relations négatives quant à elles, plus rares dans la littérature et dans les bases de connaissances, ont été générées aléatoirement, avant une vérification de leur absence dans les bases de données de référence. Deux ensembles d'apprentissage ont été générés, l'un équilibré, composé de 91 paires positives et de 91 paires négatives, l'autre déséquilibré, avec 91 paires positives et 182 paires négatives.

Une fois le modèle entraîné, nous avons classé 1,760 paires gène-médicament associées avec un faible niveau de validation selon PharmGKB, *i.e.*, des associations ayant un niveau de confiance de 3 ou 4. Parmi les 20 meilleurs candidats obtenus par les deux modèles, nous retrouvons des paires largement étudiées dans la littérature. Il est par exemple connu que les variations de l'*epidermal growth factor* (*EGFR*) sont liées à divers phénotypes cancéreux [87]. De plus, des précédentes études de Pharmacogénomique ont montré une association entre ces variations et l'efficacité de certaines molécules telles que l'Ernotinib (classé au second

rang) dans le cas du cancer du poumon à petites cellules (CPPC)[87, 88]. Cependant, nous retrouvons parmi les candidats des paires peu – ou pas – étudiées. C’est le cas notamment de l’association *MAP3K1* – Carboplatine, classée au premier rang. Une étude d’association pangénomique sur des patients atteints de CPPC et traités par Carboplatine a identifié une SNP du gène *DSCAM* comme un bon biomarqueur du pronostic [89]. Cette paire est par ailleurs retrouvée au sixième rang dans notre étude. Un possible synergie *MAP3K1*  $\times$  *DSCAM* mériterait d’être étudiée.

Malgré les résultats prometteurs de cette approche, elle présente de nombreuses limites. Tout d’abord, l’utilisation de la méthode des forêts d’arbres aléatoires s’est montrée peu adaptée aux données prises en compte. Ces dernières, modélisées sous forme de graphes, se prêtent peu à une transformation sous forme de vecteurs. Cette transformation nécessite d’effectuer des choix importants concernant les attributs à prendre en compte dans la nouvelle modélisation, pouvant être associée à une perte d’information conséquente. De plus, notre méthode de transformation conduit à une représentation en de multiples Instances d’une même relation gène–médicament. Bien que des méthodes gérant ces multiples Instances existent [90], il est préférable de garder la structure initiale des données, ce qui a été rendu possible par l’utilisation de noyaux de graphes. Une étude plus approfondie des différents paramètres de ces noyaux (profondeur du graphe, méthode de parcours, etc.) serait par ailleurs nécessaire. Enfin, le manque de données sur les relations négatives peut poser un problème majeur. L’approche prise ici, consistant à générer des paires aléatoires, peut être améliorée.

Beaucoup de travail reste donc à faire, tant dans l’enrichissement des données sources que dans les études subséquentes, permettant notamment de valider la méthode et d’expliquer les relations obtenues. Une partie de ces missions correspond à celles du projet PractiK-Pharma (Practice-based evidences for actioning Knowledge in Pharmacogenomics), financé par l’ANR. La Figure 29 présente les objectifs du projet.

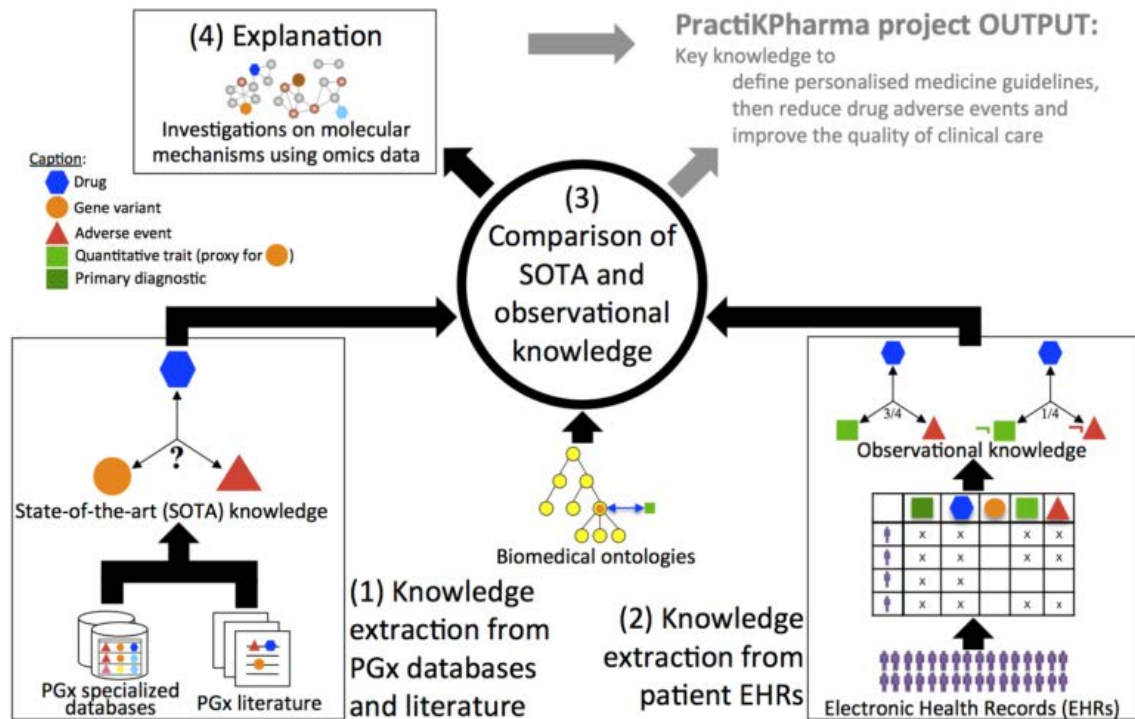


FIGURE 29 – Présentation des objectifs du projet PractiKPharma : (1) Extraire des connaissances des bases de données spécialisées et de la littérature, (2) Extraire des connaissances des dossiers médicaux, (3) De comparer les connaissances extraites de ces deux sources et enfin, (4), de tenter d'expliquer les relations nouvellement trouvées [91].

## Références

- [1] H.-G. Xie and F. W. Frueh, “Pharmacogenomics steps toward personalized medicine,” *Personalized Medicine*, vol. 2, no. 4, pp. 325–37, 2005.
- [2] Y. Garten, A. Coulet, and R. B. Altman, “Recent progress in automatically extracting information from the pharmacogenomic literature,” *Pharmacogenomics*, vol. 11, no. 10, 2010.
- [3] M. Whirl-Carrillo, E. McDonagh, J. Hebert, L. Gong, K. Sangkuhl, C. Thorn, R. Altman, and T. E. Klein, “Pharmacogenomics knowledge for personalized medicine,” *Clinical Pharmacology & Therapeutics*, vol. 92, no. 4, pp. 414–417, 2012.
- [4] J. P. Ioannidis, “To replicate or not to replicate : The case of pharmacogenetic studies,” *Circulation : Cardiovascular Genetics*, vol. 6, pp. 413–8, 2013.
- [5] I. Zineh, M. Pacanowski, and J. Woodcock, “Pharmacogenetics and coumarin dosing? Recalibrating expectations,” *N Engl J Med*, vol. 369, pp. 2273–5, 2013.
- [6] N. Hansen, S. Brunak, and R. Altman, “Generating genome-scale candidate gene lists for pharmacogenomics,” *Clinical Pharmacology & Therapeutics*, pp. 183–9, 2009.
- [7] Y. Garten, N. P. Tatonetti, and R. B. Altman, “Improving the prediction of pharmacogenes using text-derived gene-drug relationships,” in *PSB*, pp. 305–314, 2010.
- [8] M. Dumontier and N. Villanueva-Rosales, “Towards pharmacogenomics knowledge discovery with the semantic web,” *Briefings in Bioinformatics*, vol. 10, no. 2, pp. 153–163, 2009.
- [9] A. Coulet, M. Smail-Tabbone, A. Napoli, and M.-D. Devignes, “Ontology-based knowledge discovery in pharmacogenomics,” *Adv Exp Med Biol*, vol. 2, 2011.
- [10] M. Samwald, A. Coulet, I. Huerga, R. L. Powers, J. S. Luciano, R. R. Freimuth, F. Whipple, E. Pichler, E. Prud’hommeaux, M. Dumontier, and M. S. Marshall, “Semantically enabling pharmacogenomic data for the realization of personalized medicine,” *Pharmacogenomics*, vol. 13, pp. 201–212, Jan. 2012.



- [11] A. A. Mangoni and S. H. D. Jackson, “Age-related changes in pharmacokinetics and pharmacodynamics : basic principles and practical applications,” *British Journal of Clinical Pharmacology*, vol. 57, pp. 6–14, Jan. 2004.
- [12] M. J. Hanley, D. R. Abernethy, and D. J. Greenblatt, “Effect of obesity on the pharmacokinetics of drugs in humans,” *Clinical Pharmacokinetics*, vol. 49, no. 2, pp. 71–87, 2010.
- [13] F. P. Guengerich, “Role of cytochrome P450 enzymes in drug-drug interactions,” *Advances in Pharmacology*, vol. 43, pp. 7–35, 1997.
- [14] J. Lazarou, B. H. Pomeranz, and P. N. Corey, “Incidence of adverse drug reactions in hospitalized patients : a meta-analysis of prospective studies,” *Journal of the American Medical Association*, vol. 279, no. 15, pp. 1200–1205, 1998.
- [15] M. H. Court, “A pharmacogenomics primer,” *Journal of Clinical Pharmacology*, vol. 47, pp. 1087–1103, Sept. 2007.
- [16] F. Vogel, “Moderne probleme der humangenetik,” in *Ergebnisse der inneren medizin und kinderheilkunde*, pp. 52–125, Springer, 1959.
- [17] D. L. M. Ito, Ralph K., “Pharmacogenomics and pharmacogenetics : future role of molecular diagnostics in the clinical diagnostic laboratory,” *Clinical chemistry*, vol. 50, no. 9, pp. 1526–1527, 2004.
- [18] A. Coulet, *Construction et utilisation d’une base de connaissances pharmacogénomique pour l’intégration de données et la découverte de connaissances*. PhD thesis, Université Henri Poincaré-Nancy I, 2008.
- [19] J. A. Johnson, “Pharmacogenetics in clinical practice : how far have we come and where are we going?,” *Pharmacogenomics*, vol. 14, no. 7, pp. 835–843, 2013.
- [20] M. Morishita and N. A. Peppas, “Is the oral route possible for peptide and protein drug delivery?,” *Drug discovery today*, vol. 11, no. 19, pp. 905–910, 2006.

- [21] S. R. Marcsisin, G. Reichard, and B. S. Pybus, “Primaquine pharmacology in the context of CYP 2d6 pharmacogenomics : current state of the art,” *Pharmacology & therapeutics*, vol. 161, pp. 1–10, 2016.
- [22] V. G. Linga, D. B. Patchva, K. M. Mallavarapu, V. Tulasi, K. I. Kalpathi, A. Pillai, S. Gundeti, S. J. Rajappa, and R. Digumarti, “Thiopurine methyltransferase polymorphisms in children with acute lymphoblastic leukemia.,” *Indian journal of medical and paediatric oncology : official journal of Indian Society of Medical & Paediatric Oncology*, vol. 35, no. 4, pp. 276–280, 2014.
- [23] A.-C. Macherey and P. M. Dansette, “Biotransformations leading to toxic metabolites : chemical aspects,” *The Practice of Medicinal Chemistry (3rd edn)(Wermuth, CG, ed.)*, pp. 674–696, 2008.
- [24] T. Omura, “The carbon monoxide-binding pigment of liver microsomes,” *J. biol. Chem.*, vol. 239, pp. 2370–2378, 1964.
- [25] “Gene Families Index | HUGO Gene Nomenclature Committee, <http://www.genenames.org/cgi-bin/genefamilies/>, visited may 2017.”
- [26] M. T. Holmberg, A. Tornio, H. Hyvärinen, M. Neuvonen, P. J. Neuvonen, J. T. Backman, and M. Niemi, “Effect of grapefruit juice on the bioactivation of prasugrel,” *British journal of clinical pharmacology*, vol. 80, no. 1, pp. 139–145, 2015.
- [27] E. Chong and M. H. Ensom, “Pharmacogenetics of the proton pump inhibitors : a systematic review,” *Pharmacotherapy : The Journal of Human Pharmacology and Drug Therapy*, vol. 23, no. 4, pp. 460–471, 2003.
- [28] J.-S. Jang, K.-I. Cho, H.-Y. Jin, J.-S. Seo, T.-H. Yang, D.-K. Kim, D.-S. Kim, S.-H. Seol, D.-I. Kim, B.-H. Kim, and others, “Meta-analysis of cytochrome P450 2c19 polymorphism and risk of adverse clinical outcomes among coronary artery disease patients of different ethnic groups treated with clopidogrel,” *The American journal of cardiology*, vol. 110, no. 4, pp. 502–508, 2012.

- [29] T. J. Lynch, D. W. Bell, R. Sordella, S. Gurubhagavatula, R. A. Okimoto, B. W. Brannigan, P. L. Harris, S. M. Haserlat, J. G. Supko, F. G. Haluska, and others, “Activating mutations in the epidermal growth factor receptor underlying responsiveness of non-small-cell lung cancer to gefitinib,” *New England Journal of Medicine*, vol. 350, no. 21, pp. 2129–2139, 2004.
- [30] C.-H. Yun, K. E. Mengwasser, A. V. Toms, M. S. Woo, H. Greulich, K.-K. Wong, M. Meyerson, and M. J. Eck, “The T790m mutation in EGFR kinase causes drug resistance by increasing the affinity for ATP,” *Proceedings of the National Academy of Sciences*, vol. 105, no. 6, pp. 2070–2075, 2008.
- [31] K. Westbrook and V. Stearns, “Pharmacogenomics of breast cancer therapy : an update,” *Pharmacology & therapeutics*, vol. 139, no. 1, pp. 1–11, 2013.
- [32] G. Schreiber, *Knowledge engineering and management : the CommonKADS methodology*. MIT press, 2000.
- [33] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From data mining to knowledge discovery in databases,” *AI magazine*, vol. 17, no. 3, p. 37, 1996.
- [34] J. MacQueen *et al.*, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, pp. 281–297, Oakland, CA, USA., 1967.
- [35] B. Thiesson, C. Meek, D. M. Chickering, and D. Heckerman, “Learning mixtures of bayesian networks,” in *in Cooper & Moral*, Citeseer, 1997.
- [36] M. R. Anderberg, *Cluster analysis for applications : probability and mathematical statistics : a series of monographs and textbooks*, vol. 19. Academic press, 2014.
- [37] L. Kaufman and P. J. Rousseeuw, *Finding groups in data : an introduction to cluster analysis*, vol. 344. John Wiley & Sons, 2009.
- [38] J. M. Pena, J. A. Lozano, and P. Larranaga, “An empirical comparison of four initialization methods for the k-means algorithm,” *Pattern recognition letters*, vol. 20, no. 10, pp. 1027–1040, 1999.

- [39] S. Z. Selim and M. A. Ismail, “K-means-type algorithms : A generalized convergence theorem and characterization of local optimality,” *IEEE Transactions on pattern analysis and machine intelligence*, no. 1, pp. 81–87, 1984.
- [40] A. K. Jain, “Data clustering : 50 years beyond k-means,” *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.
- [41] CheCheDaWaff, “Illustration of a concave set. chechedawaf, visited may 2017. [https://fr.wikipedia.org/wiki/Fichier:Convex\\_polygon\\_illustration2.svg](https://fr.wikipedia.org/wiki/Fichier:Convex_polygon_illustration2.svg),” June 2016.
- [42] O. P. B. Yankovsky, “[https://commons.wikimedia.org/wiki/File:Dendrogram\\_of\\_the\\_genetic\\_distances\\_between\\_Russian\\_populations.svg](https://commons.wikimedia.org/wiki/File:Dendrogram_of_the_genetic_distances_between_Russian_populations.svg),” Apr. 2011.
- [43] M. M. Deza and E. Deza, *Encyclopedia of Distances*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2014.
- [44] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” in *Soviet physics doklady*, vol. 10, pp. 707–710, 1966.
- [45] R. W. Hamming, “Error detecting and error correcting codes,” *Bell Labs Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.
- [46] “<https://de.wikipedia.org/wiki/Datei:CompleteLinkage.svg>, visited may 2017.”
- [47] Sigbert, “Visualization of average linkage, <https://commons.wikimedia.org/wiki/File:AverageLinkage.svg>. visited may 2017,” Jan. 2011.
- [48] “<https://de.wikipedia.org/wiki/Datei:SingleLinkage.svg>, visited may 2017.”
- [49] “1.10. Decision Trees — scikit-learn 0.18.1 documentation.”
- [50] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*, vol. 6. Springer, 2013.
- [51] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, pp. 5–32, Oct. 2001.

- [52] J. MacCormick, “How does the kinect work, <https://users.dickinson.edu/~jmac/selected-talks/kinect.pdf>,” *Presentert ved Dickinson College*, vol. 6, 2011.
- [53] R. Genuer, *Random Forests : elements of theory, variable selection and applications*. Theses, Université Paris Sud - Paris XI, Nov. 2010.
- [54] Elisaboari, “English : SVM - Example of separating hyperplanes.”
- [55] Cyc, “Graphic showing the maximum separating hyperplane and the margin.,” Feb. 2008.
- [56] Sylenius, “separating hyperplane in 2d, to illustrate basic concept of machine learning.”
- [57] M. T. Goodrich and R. Tamassia, *Algorithm Design and Applications*. NHoboken, N.J : Wiley, 1 edition ed., Oct. 2014.
- [58] H. Singh and R. Sharma, “Role of Adjacency Matrix and Adjacency List in Graph Theory,” *International Journal of Computers & Technology*, vol. 3, no. 1, 2012.
- [59] A. Drichel, “Drichel, alexander, <https://commons.wikimedia.org/wiki/File:Breadth-first-tree.svg>, visited august 2016,” Mar. 2008.
- [60] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, “Introduction to algorithms, second edition,” 2001.
- [61] A. Drichel, “Drichel, alexander, <https://commons.wikimedia.org/wiki/File:Depth-first-tree.svg>, visited august 2016,” Mar. 2008.
- [62] E. W. Dijkstra, “A Note on Two Problems in Connexion with Graphs,” *Numer. Math.*, vol. 1, pp. 269–271, Dec. 1959.
- [63] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

- [64] N. Shervashidze, P. Schweitzer, E. J. v. Leeuwen, K. Mehlhorn, and K. M. Borgwardt, “Weisfeiler-lehman graph kernels,” *Journal of Machine Learning Research*, vol. 12, no. Sep, pp. 2539–2561, 2011.
- [65] J. R. Landis and G. G. Koch, “The measurement of observer agreement for categorical data,” *Biometrics*, vol. 33, pp. 159–174, Mar. 1977.
- [66] T. Heath and C. Bizer, “Linked Data : Evolving the Web into a Global Data Space,” *Synthesis Lectures on the Semantic Web : Theory and Technology*, vol. 1, pp. 1–136, Feb. 2011.
- [67] T. Berners-Lee, *Weaving the Web : The Original Design and Ultimate Destiny of the World Wide Web*. San Francisco : HarperBusiness, 1 ed., Nov. 2000.
- [68] “Tim Berners-Lee – Linked Data Planet 2008-06-17 (3), [https://www.w3.org/2008/Talks/0617-lod-tbl/#\(3\)](https://www.w3.org/2008/Talks/0617-lod-tbl/#(3)), visited june 2017.”
- [69] “How to publish Linked Data on the Web, <http://wifo5-03.informatik.uni-mannheim.de/bizer/pub/LinkedDataTutorial/>, visited may 2015.”
- [70] “RDF Platform < EMBL-EBI, <http://www.ebi.ac.uk/rdf/>, visited may 2015.”
- [71] A. Callahan, J. Cruz-Toledo, P. Ansell, and M. Dumontier, “Bio2rdf Release 2 : Improved Coverage, Interoperability and Provenance of Life Science Linked Data,” in *The Semantic Web : Semantics and Big Data* (P. Cimiano, O. Corcho, V. Presutti, L. Hol-link, and S. Rudolph, eds.), no. 7882 in Lecture Notes in Computer Science, pp. 200–212, Springer Berlin Heidelberg, 2013.
- [72] L. Masinter, T. Berners-Lee, and R. T. Fielding, “Uniform Resource Identifiers (URI) : Generic Syntax.”
- [73] S. L. Weibel and T. Koch, “The dublin core metadata initiative,” *D-lib magazine*, vol. 6, no. 12, pp. 1082–9873, 2000.
- [74] F. Manola, E. Miller, B. McBride, *et al.*, “Rdf primer,” *W3C recommendation*, vol. 10, no. 1-107, p. 6, 2004.

- [75] A. Callahan, J. Cruz-Toledo, P. Ansell, and M. Dumontier, “Bio2rdf release 2 : Improved coverage, interoperability and provenance of life science linked data,” in *ESWC*, 2013.
- [76] B. Percha, Y. Garten, and R. B. Altman, “Discovery and explanation of drug-drug interactions via text mining,” *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pp. 410–421, 2012.
- [77] A. Coulet, N. H. Shah, Y. Garten, M. Musen, and R. B. Altman, “Using text to build semantic networks for pharmacogenomics,” *Journal of biomedical informatics*, vol. 43, no. 6, pp. 1009–1019, 2010.
- [78] B. Percha, Y. Garten, and R. B. Altman, “Discovery and explanation of drug-drug interactions via text mining,” in *PSB*, pp. 410–421, 2012.
- [79] V. Bicer, T. Tran, and A. Gossen, “Relational kernel machines for learning from graph-structured RDF data,” in *ESWC*, pp. 47–62, 2011.
- [80] Y. Huang, V. Tresp, M. Bundschuh, A. Rettinger, and H. Kriegel, “Multivariate prediction for learning on the semantic web,” in *ILP*, pp. 92–104, 2010.
- [81] A. Thor, P. Anderson, L. Raschid, S. Navlakha, B. Saha, S. Khuller, and X.-N. Zhang, “Link Prediction for Annotation Graphs Using Graph Summarization,” in *Proceedings of the 10th International Conference on The Semantic Web - Volume Part I, ISWC’11*, (Berlin, Heidelberg), pp. 714–729, Springer-Verlag, 2011.
- [82] U. Lösch, S. Bloehdorn, and A. Rettinger, “Graph kernels for RDF data,” in *ESWC*, pp. 134–148, 2012.
- [83] G. K. D. de Vries, “A fast approximation of the weisfeiler-lehman graph kernel for RDF data,” in *ECML-PKDD*, pp. 606–621, 2013.
- [84] P. Heim, S. Lohmann, and T. Stegemann, “Interactive relationship discovery via the semantic web,” in *ESWC*, vol. 6088, pp. 303–317, 2010.
- [85] K. Dalleau, N. Coumba Ndiaye, and A. Coulet, “Suggesting valid pharmacogenes by mining linked data,” in *Semantic Web Applications and Tools for Life Sciences*

- (SWAT4LS) 2015, Proceedings of the Semantic Web Applications and Tools for Life Sciences (SWAT4LS) 2015, (Cambridge, United Kingdom), Dec. 2015.
- [86] K. Dalleau, Y. Marzougui, S. Da Silva, P. Ringot, N. C. Ndiaye, and A. Coulet, “Learning from biomedical linked data to suggest valid pharmacogenes,” *Journal of biomedical semantics*, vol. 8, no. 1, p. 16, 2017.
  - [87] C. Mayo, J. Bertran-Alamillo, M. A. Molina-Vila, A. Giménez-Capitán, C. Costa, and R. Rosell, “Pharmacogenetics of EGFR in lung cancer : perspectives and clinical applications,” *Pharmacogenomics*, vol. 13, pp. 789–802, May 2012.
  - [88] R. A. de Mello, P. Madureira, L. S. Carvalho, A. Araújo, M. O’Brien, and S. Popat, “EGFR and KRAS mutations, and ALK fusions : current developments and personalized therapies for patients with advanced non-small-cell lung cancer,” *Pharmacogenomics*, vol. 14, pp. 1765–1777, Nov. 2013.
  - [89] Y. Sato, N. Yamamoto, H. Kunitoh, Y. Ohe, H. Minami, N. M. Laird, N. Katori, Y. Saito, S. Ohnami, H. Sakamoto, J.-I. Sawada, N. Saijo, T. Yoshida, and T. Tamura, “Genome-wide association study on overall survival of advanced non-small cell lung cancer patients treated with carboplatin and paclitaxel,” *Journal of Thoracic Oncology : Official Publication of the International Association for the Study of Lung Cancer*, vol. 6, pp. 132–138, Jan. 2011.
  - [90] C. Leistner, A. Saffari, and H. Bischof, “Miforests : Multiple-instance learning with randomized trees,” *Computer Vision–ECCV 2010*, pp. 29–42, 2010.
  - [91] A. Coulet and M. Smail-Tabbone, “Mining Electronic Health Records to Validate Knowledge in Pharmacogenomics,” *ERCIM News*, p. 56, Jan. 2016.



# Glossaire

**FDA** *Food and Drug Administration*, agence étasunienne chargée, entre autres, d'autoriser la commercialisation des médicaments et des denrées alimentaires sur leur territoire. 5, 11

**KDD** *Knowledge Discovery in Databases*. Le terme KDD et le terme ECBD sont équivalents. 11

**RDF** *Resource Description Framework*, format de données standard, particulièrement adapté à l'échange de données et la représentation de ces données sous la forme d'un graphe. 12, 41–43, 45–47

**XML** *eXtensible Markup Language*, langage de balisage permettant d'encoder des documents dans un format à la fois compréhensible par la machine comme par l'humain. 12, 41, 45

**PharmGKB** Base de données contenant des relations gène–médicament manuellement annotées. 1

**Web sémantique** Ensemble de données structurées, disponibles sur le Web et interconnectées entre elles. 1, 2, 41–43, 47

**SNP** *Single Nucleotide Polymorphism*, variation d'une unique paire de bases à une position spécifique du génome. 9, 64

**ECBD** Exploration des Connaissances à partir de Bases de de Données. Il s'agit d'un processus visant l'extraction de connaissances utiles et non triviales à partir de données provenant de sources de données volumineuses. 11

**Fouille de données** Processus visant à extraire des connaissances des données disponibles. 12

**Apprentissage supervisé** Méthodes visant à apprendre une fonction à partir de données étiquetées, dans l'objectif d'inférer la classe d'objets inconnus. 20, 38, 76

**Random Forest** Méthode d'apprentissage supervisé dont le principe général est de construire une collection de prédicteurs, pour ensuite agréger l'ensemble de leurs prédictions. 24

**SVM** *Support Vector Machine* (machine à vecteur de support). Méthode d'Apprentissage supervisé permettant de classer des données séparables linéairement. 28, 29

**Graph Kernel** Fonction  $f : \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}$ , quantifiant une mesure de similarité entre deux graphes. 36

**Précision** Parmi tous les éléments prédits comme appartenant à une classe, proportion d'éléments appartenant effectivement à cette classe. 39, 40

**Rappel** Proportion d'éléments prédits comme appartenant à une classe parmi tous les éléments appartenant effectivement à cette classe. 39, 40

**F-mesure** Moyenne harmonique de la précision et du rappel. Sa valeur va de 0 pour une mauvaise classification, à 1 pour une classification parfaite. 39, 40

**Web des données** Initiative visant à favoriser la publication de données structurées sur le Web. 41

**URI** *Uniform Resource Identifier*, chaîne de caractères permettant d'identifier une ressource, présente sur un réseau (auquel cas il s'agit d'une URL, *Uniform Resource Locator* ou non, auquel cas il s'agit d'une URN, *Uniform Resource Name* . 41, 43, 44, 46

**LOD** *Linked Open Data* (données ouvertes et liées). Ensemble de données structurées, disponibles sur le Web et interconnectées entre elles. 42, 43, 47

**SPARQL** Acronyme récursif pour *SPARQL Protocol And RDF Query Language*, SPARQL étant un langage de requêtes et un protocole associé permettant d'effectuer des requêtes et de récupérer des données contenues dans des graphes RDF. 43, 45, 46, 63

**Pharmacogénomique** Étude des variations interindividuelles de réponse à un médicament, liées aux différences génétiques entre personnes ou populations. 47–49, 63

**API** *Application Programming Interface* (interface de programmation applicative). Ensemble de méthodes, fonctions et routines permettant d'accéder à des fonctions et de communiquer avec des programmes et services. 47

**Fouille de graphes** Processus de fouille de données appliquée à un type de données particuliers, les graphes. 49

**Instance** Objet/ligne d'une source de données, étiqueté ou à étiqueter. 63, 64

## DEMANDE D'IMPRIMATUR

Date de soutenance : 12/07/2017

DIPLOME D'ETAT DE DOCTEUR  
EN PHARMACIE

présenté par : Kevin Dalleau

Sujet : SUGGESTION D'INTERACTIONS GÈNE-  
MÉDICAMENT A PARTIR DE DONNÉES OUVERTES  
ET LIÉESJury :

Président : M. Luc Ferrari, Professeur des Universités

Directeur : M. Adrien Coulet, Maître de Conférences

Mme Ndeye Coumba Ndiaye, Maître de Conférences

Juges : Mme Nadine Petitpain, Praticien Hospitalier

Vu,

Nancy, le 12/06/2017

Le Président du Jury

Directeur de Thèse



M. LUC FERRARI

M. ADRIEN COULET

MME NDEYE COUMBA  
NDIAYE

Vu et approuvé,

Nancy, le 16.06.2017

Doyen de la Faculté de Pharmacie  
de l'Université de Lorraine,  
Francine PAULUS  


Vu,

Nancy, le 22.06.2017

Le Président de l'Université de Lorraine,



Pierre MUTZENHARDT

N° d'enregistrement : 9911

**TITRE**

**SUGGESTION D'INTERACTIONS GÈNE-MÉDICAMENT A PARTIR DE DONNÉES OUVERTES ET LIÉES**

Thèse soutenue le 12/07/2017

Par Kevin Dalleau

**RESUME :**

La pharmacogénomique étudie comment les variations génétiques inter-individuelles impactent la réponse aux médicaments. Une tâche courante pour les chercheurs du domaine consiste à identifier les gènes pouvant intervenir dans la variabilité de la réponse entre individus, les pharmacogènes. Il s'agit d'une tâche complexe, de par la rareté de certains variants, la difficulté de récolter des données fiables, ou encore la complexité d'effectuer des études robustes. Pour répondre à ces problématiques, les approches computationnelles comme celles proposées dans cette thèse constituent des alternatives prometteuses.

Nous proposons dans ce travail de sélectionner diverses sources de données biomédicales d'intérêt en pharmacogénomique, et d'y appliquer des méthodes de fouille afin d'identifier et de proposer des pharmacogènes valides. Nous avons fait le choix de nous focaliser sur six sources de données que nous avons mis en forme selon les principes des données ouvertes et liées lorsqu'elles n'étaient pas déjà disponibles ainsi. L'utilisation des données ouvertes et liées rend possible l'utilisation conjointe de sources multiples, étant construites et distribuées selon un framework commun. Deux méthodes ont été expérimentées et comparées dans ce travail afin de classer des paires gènes-médicaments : les forêts d'arbres aléatoires et les graph kernels.

Après avoir fusionné les sources sélectionnées, nous avons obtenu un graphe de 2 610 793 triplets. Bien que l'apprentissage sur ces données en utilisant les forêts d'arbres aléatoires nous donnent un résultat satisfaisant (F-mesure = 0.73), l'application des graph kernels a donné les meilleurs résultats (F-mesure = 0.81). Nous proposons à l'issue de ce travail une liste de pharmacogènes candidats issus des deux approches, ainsi qu'une discussion concernant leur présence dans cette liste.

**MOTS CLES :**

PHARMACOGÉNOMIQUE, INTERACTIONS GÈNE-MÉDICAMENT, WEB DES DONNÉES, LINKED OPEN DATA, RANDOM FORESTS, KNOWLEDGE DISCOVERY, BIO2RDF, BASES DE DONNÉES

Directeur de thèse	Intitulé du laboratoire	Nature
<u>Mr. Adrien Coulet</u>	<u>LORIA, UMR 7503</u>	Expérimentale <input checked="" type="checkbox"/>
<u>Mme Ndeye-Coumba Ndiaye</u>	<u>UMR INSERM U1122 IGE-PCV</u>	Bibliographique <input type="checkbox"/>
		Thème 1

**Thèmes**

1 – Sciences fondamentales	2 – Hygiène/Environnement
3 – Médicament	4 – Alimentation – Nutrition
5 – Biologie	6 – Pratique professionnelle