Construction et utilisation d'une base de connaissances pharmacogénomique pour l'intégration de données et la découverte de connaissances

Adrien Coulet

Département de formation doctorale en informatique **École doctorale IAEM Lorraine**
UFR STMIA

Construction et utilisation d'une Base de Connaissances pharmacogénomique pour l'intégration de données et la découverte de connaissances

THÈSE

présentée et soutenue publiquement le 10 octobre 2008

pour l'obtention du

Doctorat de l'université Henri Poincaré – Nancy 1

(spécialité informatique)

par

Adrien Coulet

Composition du jury

Rapporteurs :	Mohand-Saïd Hacid	Professeur, Université Claude Bernard, Lyon 1
	Alain Viari	Directeur de Recherche, INRIA
Examinateurs :	Nacer Boudjlida	Professeur, Nancy Universités
	Marie-Dominique Devignes	Chargée de Recherche, CNRS
	Chantal Reynaud	Professeur, Université Paris-Sud 11
	Malika Smaïl-Tabbone	Maître de conférence, Nancy Universités
Invités :	Pascale Benlian	Maître de conférence - Praticien hospitalier, Université Pierre et Marie Curie, Paris 6
	Amedeo Napoli	Directeur de Recherche, CNRS

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thloria.

Remerciements

Table des matières

Table des figures

Liste des tableaux

Introduction

Ce chapitre est une introduction générale à la thèse. Les sections 1–3 décrivent le contexte scientifique de la thèse : la découverte de connaissances pour la pharmacogénomique. La section 4 présente le contexte industriel qui a motivé cette thèse. La section 5 introduit sa problématique, puis la section 6 l'approche adoptée et les principales contributions. Enfin, la section 7 présente un résumé des chapitres de la thèse.

1 Des données aux connaissances

L'expansion du nombre de sources de données disponibles en particulier grâce au Web et la quantité de données gérées au sein de ces sources ont rendu indispensable la mise au point de systèmes capables d'extraire de façon automatique, ou semi-automatique, des connaissances disponibles mais cachées par la complexité des données. Cette complexité est principalement due à l'hétérogénéité, la diversité, la dispersion, et le grand volume des données. Le processus d'Extraction de Connaissances à partir de Bases de Données (ECBD), décrit par Frawley *et al.* [FPSM91], a justement pour but la découverte d'unités de connaissances à partir d'ensembles de bases de données volumineuses.

Avant de définir et détailler le processus d'ECBD, il convient de préciser la distinction que nous faisons dans cette thèse entre données, information, et connaissances. De nombreuses tentatives de définition ont vu le jour notamment dans le domaine des sciences cognitives où l'exploitation d'informations diverses par un système complexe permet l'acquisition de connaissances capables de diriger la mise en œuvre d'actions. Nous nous limiterons aux définitions acceptées de manière générale dans le domaine de l'informatique exprimées par Kayser de la façon suivante [Kay97] :

- les données sont le résultat d'observations,
- les informations sont le résultat de l'interprétation de ces données,
- les connaissances définissent la façon d'utiliser les données et informations.

Cette distinction est présentée de façon plus formelle par Devlin, Schreiber, et Wille [Dev99, SAA+99, Wil02] de la façon suivante :

- données = signes + syntaxe,
- information = données + sens (ou sémantique),
- connaissances = information assimilée et interprétée + possibilité de mise en action de l'information interprétée.

Prenons un exemple relevant du domaine de la génétique et considérons la séquence d'ADN constitutive d'un gène au cœur d'une cellule. A ce niveau, la séquence de nucléotides, *i.e.* l'enchaînement de plusieurs milliers d'A, C, G, et T, peut être considérée comme des données brutes. En revanche, le fait

que l'on sache que cette séquence est reconnue par la machinerie cellulaire comme un gène particulier est une information. Enfin, les règles de fontionnement de la machinerie cellulaire et particulièrement le code génétique de la cellule constituent les connaissances qui permettent d'interpréter ce gène comme une protéine, utilisée ensuite dans la mise en œuvre de fonctions biologiques.

Dans un ordinateur les données, informations, et connaissances peuvent être représentées selon les formes suivantes :

- données : un nombre, une image, une chaîne de caractères, par exemple "ATCGGCTAGCTTATATC-GATCGAT" ;
- information : des données dans une base de données ou sous forme de tableau, associées aux métadonnées nécessaires à leur interprétation : souvent sous la forme d'un couple attribut–valeur comme par exemple "sequence_du_gene = ATCGGCTAGCTTATATCGATCGAT" ;
- connaissances : des contraintes, des règles, des axiomes logiques utilisables par des programmes pour exploiter les informations dans le cadre de la réalisation d'une action : par exemple l'aide à la décision, le pilotage d'un robot, la découverte de nouvelles connaissances.

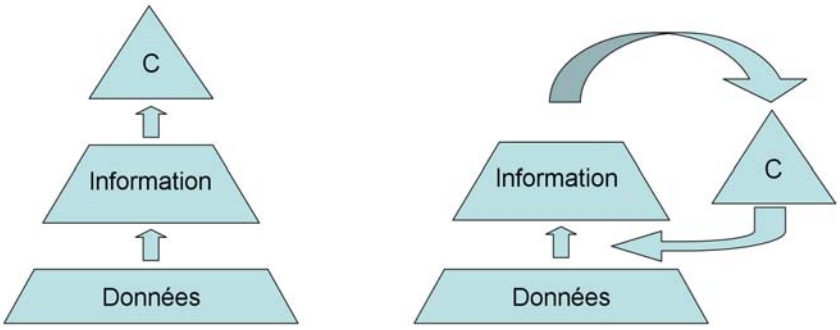

FIG. 1 – Représentation schématique et naïve du processus de transformation de données en information puis en connaissances. A gauche un processus en pyramide et à droite en boucle. La lettre C représente les connaissances.

D'un point de vue opérationnel, il est intéressant d'envisager les processus qui permettent de passer de données à l'information puis aux connaissances. De façon naïve ce processus peut être représenté sous forme de pyramide où les connaissances occupent la place la plus haute pour souligner le fait que de nombreuses données sont nécessaires à l'acquisition d'une connaissance [SAA+99, Wil02]. Il nous semble plus exact de proposer un schéma composé d'une boucle, dans la mesure où les connaissances existantes peuvent servir pour associer un sens (*i.e.* une sémantique) à des données (voir Figure 1).

Nous remarquerons toutefois qu'en informatique, la distinction données–information est artificielle puisque les programmes ne manipulent que des données (le nom d'un attribut ou une valeur attribuée à celui-ci). Comme observe Guus Schreiber dans son livre sur la méthodologie de gestion des connaissances *CommonKADS*, que ce soit pour un programme ou un humain, la frontière entre données et information n'est pas franche car elle est fortement dépendante du contexte d'utilisation [SAA+99]. Ainsi le sens associé à une donnée peut être différent d'un pays à l'autre, d'un domaine professionnel à l'autre. De même des données peuvent être chargées de sens pour un utilisateur averti et à ce titre constituer une source d'information alors qu'elles n'auront aucun sens et resteront au stade de données pour un utilisateur non averti.

Les connaissances constituent une notion nettement distincte de celles de données et d'information

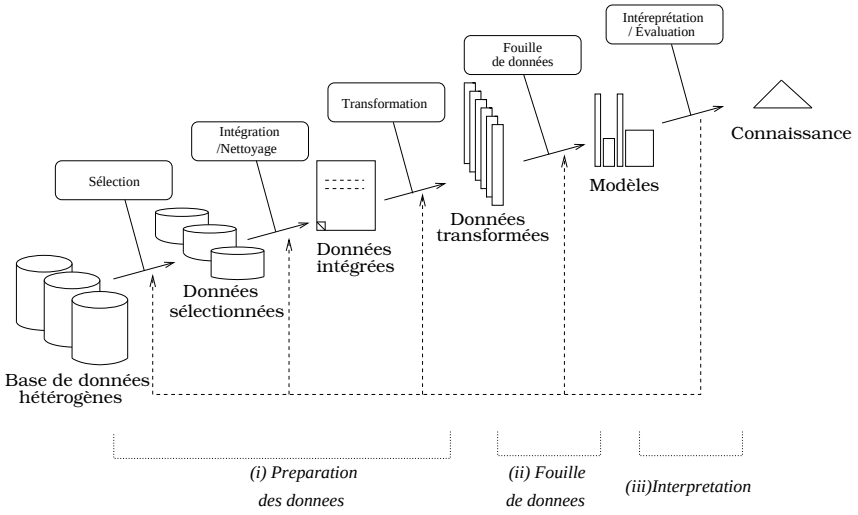

FIG. 2 – La représentation classique du processus d'Extraction de Connaissances à partir des Bases de Données (ECBD) (d'après [FPSS96])

qui restent purement descriptives. De façon différente, les connaissances se distinguent par leur caractère dynamique, orientées vers l'action comme par exemple la prise de décision ou l'acquisition de nouvelles connaissances. Ainsi la représentation des connaissances en informatique est toujours associée à des mécanismes de raisonnement qui permettent la résolution de problèmes.

Dans cette thèse nous nous intéressons particulièrement au processus d'ECBD. Celui-ci a justement pour but la découverte d'unités d'information (ou unités extraites) à partir d'ensembles de bases de données volumineuses. Ces unités d'information pourront ensuite être interprétées comme des unités de connaissance non triviales, potentiellement utiles, et réutilisables. Généralement, le processus d'ECBD est appliqué à la fois de façon *itérative* et *interactive*. Itérative car les résultats produits peuvent être réutilisés lors des itérations suivantes du processus. Interactive car le processus d'ECBD est réalisé sous le contrôle d'un expert du domaine étudié : l'analyste. C'est lui qui guide le processus en fonction de ses objectifs, de ses propres connaissances du domaine, et des résultats obtenus lors des précédentes itérations de l'extraction.

Nous distinguons trois étapes principales dans le processus, représentées Figure 2 :

– *(i)* la préparation des données incluant leur sélection, leur intégration, et leur nettoyage en vue de leur utilisation par les algorithmes de fouille de données,
– *(ii)* l'opération de fouille de données proprement dite conduisant à l'extraction d'unités d'information présentes sous forme de régularités dans les données, et
– *(iii)* l'interprétation des unités d'information extraites en terme de connaissance.

Les unités de connaissance ainsi produites peuvent être exprimées dans un formalisme de représentation des connaissances afin de pouvoir être utilisées dans des systèmes fondés sur les connaissances.

Dans la suite de cette thèse nous ne considérerons pas la notion d'information très dépendante de l'interprétation individuelle. Plutôt que d'employer le terme d'unité d'information, nous préférerons parler d'*unités extraites* par la fouille de données qui peuvent revêtir différentes formes selon l'algorithme de fouille utilisé : un motif fréquent, un concept formel, une règle d'association, un cluster, etc. En revanche nous nous attacherons à étudier ce qui distingue les données des connaissances : le couple <syntaxe, sémantique formelle> et comment ce couple est exploité par des mécanismes de raisonnement pour mettre en action les connaissances.

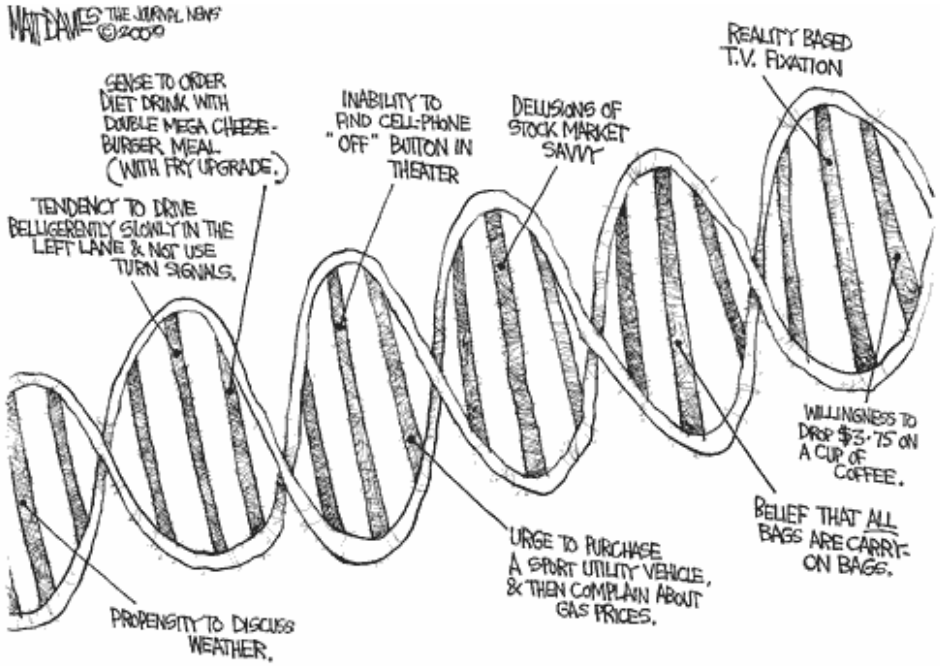

FIG. 3 – Une annotation humoristique du génome humain par Matt Davies. *The Journal News/Los Angeles Times* Syndicate, 2000.

Dans le domaine de la génomique et plus généralement de la biologie moléculaire, les progrès biotechnologiques ont mené à l'acquisition de larges volumes de données puis à leur enregistrement dans des centaines de bases de données créées spécialement [Bat08]. Par exemple, les techniques de séquençage moléculaire de l'ADN ont permis le décryptage d'une première version du génome humain en 2001 mis ensuite à la disposition de la communauté scientifique dans trois bases de données [ea01, Con01] :

- Genome Browser[1] de l'UCSD (*University of California Santa Cruz*),
- Ensembl[2] de l'EBI (*European Bioinformatics Institute*), et
- Map Viewer[3] du NCBI (*National Center for Biotechnology Information*).

D'autres projets, depuis, s'intéressent à affiner la séquence du génome et à l'annoter (*i.e.* la caractériser) en explorant entre autres les variations génomiques, le transcriptome, le protéome. La Figure 3 représente de façon humoristique des annotations du génome humain. De façon plus sérieuse, le projet international HapMap, terminé en 2007, et le projet *1000 genomes*, initié en 2008, explorent les variations inter-individuelles du génome humain avec entre autres l'objectif d'enrichir son annotation [Con03, Spe08]. La somme de données collectées est particulièrement intéressante en génomique fonctionnelle ou en génomique intégrative qui étudie l'impact, sur le fonctionnement du génome, de facteurs environnementaux comme l'alimentation, un traitement médicamenteux, ou des habitudes de vie.

Cependant, les médecins et les biologistes qui utilisent quotidiennement ces bases de données dans leur activité de diagnostic et de recherche sont limités par la complexité des données. Premièrement, le nombre et la dispersion des sources compliquent les tâches de collecte manuelle de données. Secondement, le volume, ainsi que des considérations plus spécifiques aux sciences du vivant comme la grande

[1] `http://genome.ucsc.edu/cgi-bin/hgGateway`
[2] `http://www.ensembl.org/Homo_sapiens/index.html`
[3] `http://www.ncbi.nlm.nih.gov/projects/mapview/`

variété des données, leur tendance à être fortement interconnectées et leurs références à des domaines spécialisés compliquent l'analyse et l'interprétation.

Face à cette difficulté, l'ECBD propose un cadre méthodologique qui a été appliqué avec succès en biologie pour : *intégrer* les données représentées dans des formats hétérogènes et dispersées dans différentes sources [GS08] et *analyser* les données par des méthodes de fouille afin d'en extraire des régularités (ou des irrégularités) [WZTS05].

Cependant, rares sont les travaux qui réutilisent effectivement les connaissances extraites ou qui tirent parti des connaissances déjà existantes pour faire face à la complexité des données post-génomiques.

2 Des connaissances pour de nouvelles connaissances

Un axe de recherche de l'équipe-projet INRIA Orpailleur est de guider le processus d'ECBD, non plus seulement par les connaissances de l'analyste, mais également par des connaissances exprimées dans un langage de représentation des connaissances particulier, interprétable par une machine [LNST08]. Le processus d'ECBD ainsi guidé par les connaissances du domaine est appelé ECCD pour *Extraction de Connaissance guidée par les Connaissances du Domaine* (ou KDDK en anglais pour *Knowledge Discovery guided by Domain Knowledge*).

De nombreux travaux en intelligence artificielle se sont intéressés à la représentation formelle de connaissances dans l'objectif de rendre celles-ci interprétables aussi bien par une machine que par un être humain. C'est notamment l'objectif du Web sémantique tel que le décrit Tim Berners-Lee [BLHL01] de proposer une extension du Web actuel dans laquelle les machines "comprennent" les informations auxquelles elles accèdent et sont ainsi en mesure de les manipuler en tant que connaissances, au sein de mécanismes de raisonnement automatiques.

A la base de l'infrastructure d'applications fondées sur les connaissances comme le Web sémantique, se trouvent les *ontologies*. Le terme ontologie fait référence à diverses notions connexes : branche de la philosophie, vocabulaire contrôlé, taxonomie, ordre partiel par exemple. Aussi la définition adoptée dans cette thèse est celle de Thomas Gruber, qui vaut pour les ontologies des applications fondées sur les connaissances, selon laquelle une ontologie est une description formelle des concepts relatifs à un domaine et des relations entre ces concepts [Gru93].

Le Web Sémantique et l'effervescence qu'il suscite ont mené la communauté scientifique au développement de standards, notamment pour la représentation des connaissances. Le langage *OWL* (*Web Ontology Language*) est ainsi le langage standard pour la représentation des ontologies du Web Sémantique. OWL est issu à la fois des langages du Web (HTML, XML, RDF) et de formalismes logiques, tels que les logiques de descriptions.

Des centaines d'ontologies exprimées en OWL sont partagées publiquement via le Web. En bioinformatique, le besoin de modélisation et d'interopérabilité des modèles biologiques, en particulier pour rendre possible l'intégration de données, a favorisé le partage et le développement communautaire de *bio-ontologies* via des portails Web comme le Bioportal ou l'OBO-Foundry [RMKM08, SAR+07].

Il est établi que les méthodes de représentation des connaissances constituent un atout pour participer au décryptage des masses de données collectées en sciences du vivant, en grande partie car elles permettent la modélisation de leur diversité et de leur hétérogénéité [Rec00, Ste08]. Les applications RiboWeb et EcoCyc illustrent notamment comment des bio-ontologies peuvent être utilisées pour favoriser l'exploitation de données biologiques [ABC+99, KACV+04]. Le langage OWL comme standard et les portails comme zone de partage et de structuration des connaissances en sciences du vivant sont deux avancées qui doivent favoriser le succès des approches fondées sur les connaissances pour la découverte de connaissances en biologie.

Ainsi, l'objectif général de cette thèse est d'étudier comment les connaissances formalisées dans

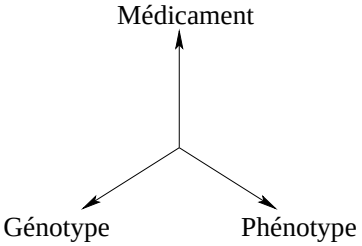

FIG. 4 – Représentation schématique de la relation entre médicament, génotype, phénotype étudiée en pharmacogénomique

une ontologie peuvent faciliter le processus de découverte de connaissances et cela notamment dans le domaine de la pharmacogénomique.

3 La pharmacogénomique

La *pharmacogénomique* étudie comment certaines variations génétiques inter-individuelles impliquent une variabilité dans les réponses entre individus à un même traitement médicamenteux [Web97].

La distinction classique entre la *pharmacogénétique* et la pharmacogénomique présente la pharmacogénétique comme l'étude des caractères héréditaires associés à la pharmacologie et la pharmacogénomique comme l'étude au niveau moléculaire de ces facteurs génétiques, de leurs interactions mutuelles, et de leurs conséquences multiples tant au niveau macroscopique qu'au niveau microscopique (moléculaire, cellulaire, tissulaire). Ainsi une définition plus complète de la pharmacogénomique comprend l'étude de l'ensemble des gènes ayant une influence sur la pharmacologie, des manifestations de leurs variations génomiques , de l'interaction de ces variations dans la production de phénotypes, et de l'influence d'un tel phénotype sur la réponse à un médicament [AK02].

La pharmacogénomique peut être schématiquement représentée comme l'étude des relations ternaires existant entre un *traitement médicamenteux*, un *génotype*, et un *phénotype* (Figure 4).

- Typiquement, le traitement médicamenteux correspond à l'administration d'une (ou plusieurs) molécule(s) avec une certaine dose, une certaine fréquence et via une certaine voie d'administration.
- Le génotype correspond à une (ou plusieurs) version(s) d'une variation génomique. Le plus souvent il s'agit du génotype (*i.e.* deux allèles pour les espèces diploïdes) observé sur le site d'une variation ponctuelle du génome, *i.e.* un *Single Nucleotide Polymorphism (SNP)*.
- Le phénotype distingue généralement trois classes qui correspondent à trois types de réponses au médicament : la réponse attendue, l'absence d'effet, une réponse adverse au médicament.

Un exemple d'interaction pharmacogénomique décrite par Desmeules *et al.* [DGDM91] et Gasche *et al.* [GDF+04] est l'influence des variations du gène *CYP2D6* dans la réponse à un traitement de codéine. La codéine est un opiacé prescrit, entre autres, pour son pouvoir analgésique. La codéine est physiologiquement métabolisée dans le foie en morphine, responsable de son effet analgésique. Il existe plusieurs versions fonctionnelles du gène *CYP2D6* (*i.e.* plusieurs variants du gène) dont les produits agissent différemment sur la transformation de codéine en morphine et permettent de distinguer plusieurs catégories d'individus (Figure 5) : les métaboliseurs lents (porteurs de variants à *activité faible*), les métaboliseurs rapides (porteurs de variants à *activité normale* ou *forte*), les métaboliseurs ultra-rapides (porteurs de copies multiples de variants à *activité normale* ou *forte*). Les métaboliseurs lents sont incapables de métaboliser efficacement la codéine en morphine, et en conséquence ne présentent pas l'effet analgésique attendu. Les métaboliseurs ultra-rapides métabolisent la codéine avec une efficacité accrue

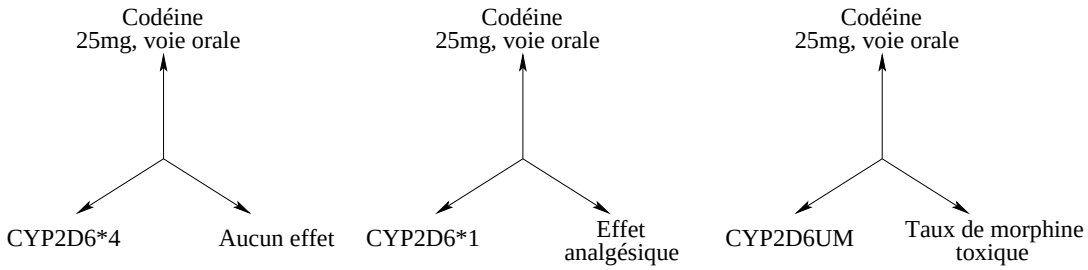

FIG. 5 – Trois exemples de relations pharmacogénomiques particulières entre un traitement de codéine, la version du gène *CYP2D6* et l'effet de la codéine. En fonction de la version du gène la réponse est différente. De gauche à droite, le cas de métaboliseurs lents, normaux ou légèrement rapides, et ultra-rapides. Il est intéressant de noter que l'administration d'une dose de codéine plus élevée (50 mg par exemple) aux métaboliseurs lents permet de compenser la limitation de l'activité enzymatique et d'obtenir l'effet analgésique attendu.

qui entraîne une intoxication à la morphine.

Les résultats des projets de collecte de données à haut débit comme le séquençage du génome, de ses variations, l'étude des transcriptome et protéome alimentent le développement de la pharmacogénomique. Le bénéfice des méthodes développées et des connaissances ainsi acquises constitue un catalyseur pour les chercheurs en biologie médicale qui voient là une occasion de bénéficier des découvertes en biologie moléculaire pour obtenir des résultats en pratique clinique [ER99]. Ce type d'importation des découvertes "théoriques" pour le monde clinique s'inscrit dans un effort général de recherche biomédicale appelé la *médecine translationelle* (traduit directement de l'anglais *translational medicine*) [Mar03]. Il est intéressant de noter que la recherche translationelle s'intéresse également au cheminement inverse, c'est à dire à étudier comment des découvertes et des pratiques cliniques peuvent être utiles pour progresser en biologie moléculaire.

Une application attendue de la pharmacogénomique au niveau des pratiques cliniques est la généralisation des prescriptions médicales individualisées, prenant en considération une exploration préalable du génotype du patient. Ceci permettrait d'optimiser l'efficacité du traitement et d'en prévenir les réponses adverses [ER04]. Cette application, appelée *médecine individualisée* (*individualized medicine* an anglais) intéresse les pouvoirs publics qui y voient un intérêt économique évident. La pharmacogénomique intéresse également les industries pharmaceutiques dans la mesure où les variations inter-individuelles dans les réponses aux médicaments peuvent expliquer pourquoi des molécules efficaces sur un panel restreint, s'avèrent, après de coûteux développements, inefficaces (voire dangereuses) au sein d'une population plus vaste. C'est justement le cas du BiDil, un médicament développé pour réduire le risque d'accidents cardio-vasculaires. Le BiDil s'était révélé inefficace lors des phases finales de son programme de mise sur le marché, puis après avoir été mis quelques temps de côté, il a été montré qu'il était particulièrement efficace pour un groupe particulier de population : les afro-américains [TZY+04]. Cet exemple alimente par ailleurs un débat éthique sur les dérives qui pourraient être associées au développement et à la prescription de molécules réservées à des sous-groupes de populations [SHSD08].

En conclusion, la pharmacogénomique est un domaine qui présente un fort intérêt médical et qui bénéficie de bases de données biologiques et de méthodes informatiques innovantes [AK02]. Ces deux arguments font de la pharmacogénomique un domaine d'application pertinent pour ce travail de thèse.

4 Le projet GenNet

Le travail présenté dans cette thèse a été initié dans le cadre d'un projet européen EUREKA, intitulé *GenNet*, impliquant les sociétés KIKA Medical, Phenosystems et l'équipe Orpailleur du LORIA.

L'idée d'origine du projet est basée sur le constat qu'un nombre grandissant d'essais cliniques inclut parmi les variables explorées des données génétiques : les résultats de génotypage de marqueurs biologiques chez les patients étudiés. Le projet GenNet se proposait de développer une infrastructure qui dans ce contexte soit capable

(1) d'enregistrer conjointement données *cliniques classiques* (pression artérielle, mesure du cholestérol total dans le sang, etc.) et données *génétiques* (ici le génotype de variations génomiques) relatives à un groupe de patients, et

(2) d'analyser les relations entre les variables ainsi collectées.

Dans ce contexte, un sujet de thèse a été proposé pour explorer deux problèmes connexes au projet industriel :

(a) compléter les données de l'essai clinique avec des données issues de bases de données biologiques publiques ou privées. Ces nouvelles données constituent un ensemble d'annotations[4] issues des travaux de recherche en biologie moléculaire qui peuvent s'avérer utiles dans l'analyse des données relatives à la population étudiée. Ces annotations supplémentaires peuvent par exemple permettre de mieux caractériser un sous-groupe de patients.

(b) proposer une approche d'analyse originale qui utilise les connaissances du domaine pour faire face à la complexité spécifique des données biologiques en terme d'hétérogénéité, de variété, de spécificité et en extraire des connaissances potentiellement utiles.

Il est possible d'imaginer que des résultats ainsi obtenus puissent, à leur tour, être à la base de travaux en biologie moléculaire et ainsi boucler la boucle de la médecine translationelle décrite en section 3 de cette introduction.

5 Problématique

Pour extraire des connaissances à partir de données hétérogènes et s'aider pour cela des connaissances existantes, notamment dans le domaine de la pharmacogénomique, nous nous sommes interessés à deux problèmes principaux : le premier consiste en la réconciliation indispensable des différentes données selon une syntaxe et une sémantique commune, le second consiste à étendre les méthodes d'extraction de connaissances pour leur permettre non seulement de travailler avec des données, mais également avec une sémantique associée aux données.

Le premier problème se pose lorsque l'on souhaite intégrer des données provenant de sources aux schémas distincts. Il est dans ce cas nécessaire de déterminer des correspondances entre les entités équivalentes représentées dans les différents schémas. Ce problème est accentué par le fait que, souvent, la sémantique associée aux entités représentées à travers les schémas des sources de données n'est pas énoncée clairement. Par exemple, le nom d'un attribut et les valeurs qu'il prend ne suffisent pas à déterminer précisement ce que représente le couple attribut–valeur, et laisse ainsi une part de liberté à l'interprétation de l'utilisateur. Une sémantique précise peut être associée aux attributs et à leurs valeurs à l'aide de descriptions formelles établies dans une ontologie. Une telle association nécessite la *mise en*

[4] De façon très générale, une annotation est une données associée à une séquence constitutive du génome pour permettre son interprétation par des biologistes.

correspondance non triviale d'une part des schémas des sources de données et d'autre part des concepts et relations définies dans une ontologie.

Le second problème se pose lorsque l'on souhaite utiliser des connaissances de domaine pour guider l'extraction de connaissances. En effet, même lorsque les schémas de données sont associés aux concepts d'une ontologie, les algorithmes de fouille de données au coeur du processus, ne sont pas capables de prendre en considération cette association. De plus, si les unités extraites par la fouille sont de manière assez naturelle réutilisées par l'analyste lors des itérations successives du processus d'ECBD, il est plus rare que le soient des unités de connaissances validées et potentiellement formalisées selon une sémantique précise. Cela nécessiterait soit l'adaptation des diverses étapes du processus d'ECBD pour qu'à chaque étape les connaissances disponibles puissent être réutilisées, soit de proposer des méthodes alternatives capables de prendre en entrée des données et des connaissances préalablement mises en correspondance.

6 Approche et principales contributions

Pour traiter les deux problèmes dégagés dans la section précédente nous proposons dans cette thèse une approche centrée sur une *Base de Connaissances* (ou BC), *i.e.* une ontologie à laquelle est associée la description d'un état particulier du domaine représenté sous la forme d'*assertions*. La première partie de cette approche consiste à réconcilier des données contenues dans des sources hétérogènes en les exprimant selon les termes de l'ontologie. La seconde partie de l'approche consiste à extraire de nouvelles connaissances de la Base de Connaissances en exploitant conjointement les régularités présentes dans les données (et conservées dans la Base de Connaissances) et les connaissances du domaines declarées explicitement dans cette base.

Intégration de données *guidée par* une ontologie Nous avons construit deux ontologies en OWL relatives aux domaines des variations génomiques et de la pharmacogénomique suivant une méthodologie rigoureuse. Nous proposons une approche d'intégration de données qui exploite ces ontologies originales pour guider l'intégration des données relatives à ces domaines. Les données et leurs schémas sont utilisés pour peupler les Bases de Connaissances associées aux ontologies à l'aide de mises en correspondance et de fonctions de transformation entre données et connaissances. Les Bases de Connaissances résultantes offrent une vue indirecte mais homogène sur l'ensemble de ces données et nous a permis, entre autres, d'évaluer le taux de recouvrement des sources intégrées.

Extraction de connaissances *à partir* d'une Base de Connaissances Nous proposons une méthode originale d'extraction de connaissances qui utilise la sémantique associée aux instances d'une Base de Connaissances obtenue suivant l'approche d'intégration décrite ci-dessus. Cette méthode appellée Analyse des Assertions de Rôles s'attache à explorer les régularités dans les assertions d'une Base de Connaissances. Les unités de connaissances produites sont exprimées suivant le même formalisme que celui de la Base de Connaissances ce qui permet, ensuite, leur manipulation par des mécanismes de raisonnement pour leur intégration cohérente à l'ensemble des connaissances préalables.

7 Plan du manuscrit

Ce manuscrit est organisé en 4 chapitres. Les deux premiers fixent le contexte biologique et l'état de l'art relatifs à la problématique de la thèse. Les deux suivants présentent les contributions de la thèse. La dernière partie est une conclusion du travail.

Chapitre 1 : Contexte biologique et applicatif Dans ce chapitre nous présentons les notions biologiques utilisés dans la thèse : les relations génotype–phénotype, les variations génomiques, et la pharmacogénomique.

Chapitre 2 : Etat de l'art Ce chapitre présente le processus d'Extraction de Connaissances *à partir* de Bases de Données (ECBD) puis deux systèmes de représentation des connaissances en rapport avec les contributions de cette thèse. Il illustre ensuite comment une représentation des connaissances peut être utilisée pour guider l'extraction de connaissances tout d'abord lors de l'étape d'intégration de données puis plus généralement lors de chacune des étapes du processus d'extraction de connaissances.

Chapitre 3 : Ontologies pour l'intégration de données en pharmacogénomique Ce chapitre présente la première contribution, à savoir l'utilisation d'ontologies originales, construites dans le cadre de la thèse, pour l'intégration de données pharmacogénomiques. Il est donc dédié premièrement à la présentation de nos ontologies SNP-Ontology et SO-Pharm et à la méthode rigoureuse mise en œuvre pour les construire. Deuxièmement, il décrit l'approche proposée pour intégrer des données à l'aide de ces ontologies. Troisièmement, sa dernière section présente les expérimentations menées dans le cadre de l'intégration de données relatives aux variations génomiques et à la pharmacogénomique.

Chapitre 4 : Extraction de connaissances dans le contexte d'une Base de Connaissances Ce chapitre détaille les deuxième et troisième contributions de la thèse, à savoir deux utilisations originales de bases de connaissances pour guider l'extraction de connaissances. La première se concentre sur l'étape de sélection des données à considérer et est illustrée par des scénarios de recherche de relations génotype–phénotype. La seconde, quant à elle, décrit la méthode d'Analyse des Assertions de Rôles. Nous proposons par cette méthode d'extraire des connaissances *à partir* d'une Base de Connaissances. Une expérimentation sur l'extraction de connaissances à partir de connaissances en pharmacogénomique termine ce chapitre.

Conclusion et perspectives Cette dernière partie conclut ce travail et en dégage les perspectives.

Chapitre 1

Contexte biologique et applicatif

Ce chapitre est une introduction aux notions de biologie abordées dans cette thèse. La première section définit les notions de génotype et de phénotype et introduit les relations existant entre ces deux notions et l'intérêt de les étudier. La deuxième section présente les variations génomiques qui consistent en des variations de la composition moléculaire du génotype et qui peuvent expliquer des modifications du phénotype. La troisième et dernière section de ce chapitre présente la pharmacogénomique, domaine d'application de ce travail de thèse. Les problématiques biologiques propres à ce domaine motivent notamment de larges parties de ce travail.

1 Génotype et phénotype

1.1 Définitions

Le *génotype* d'un individu est l'ensemble des données portées par le *génome* de cet individu, en d'autres termes l'ensemble de son matériel génétique. Pour la plupart des organismes, ce matériel génétique est codé sous forme de séquences d'*Acide Désoxyribonucléique* ou *ADN* composées par l'enchaînement de quatre molécules particulières : les *nucléotides*, notés A, C, G, et T (abréviations de leurs noms complets *Adénine*, *Cytosine*, *Guanine*, et *Thymine*). Chez l'homme et les eucaryotes en général, l'ADN est porté par les chromosomes, eux même situés dans le noyau de chaque cellule. Le génotype constitue les données de bases exploitées par les cellules pour définir les caractères d'un individu.

Le *phénotype* est, quant à lui, l'ensemble des traits observables d'un individu et résulte de l'expression de son génotype. Il est important de préciser que l'expression du génotype, et donc le phénotype qui en résulte, sont sensibles à l'influence de facteurs multiples : le moment de la vie, l'environnement, la nutrition, le stress, la maladie ou un traitement médicamenteux.

1.2 Transcription et traduction : premières étapes de la définition d'un phénotype

L'expression du génotype en un ensemble de traits observables se fait suivant deux étapes principales : la *transcription* et la *traduction* représentées Figure 1.1 de façon simple. L'unité fonctionnelle du génome considérée par la cellule lors de la transcription est le *gène*, qui est donc délimité sur l'ADN par des signaux de début et de fin de transcription. Aussi, suivant cette première étape, un gène est transcrit, c'est à dire converti, en une séquence de nucléotides dont la composition diffère légèrement de celle de l'ADN par le fait que les nucléotides T (les Thymines) sont transcrits en nucléotides U (abréviation d'*Uracile*). Cette nouvelle molécule, appelée ARN_m pour Acide Ribonucléique Messager, peut sortir du noyau de la cellule où reste l'ADN, pour ensuite subir une nouvelle transformation : la traduction.

L'ARN_m est traduit selon la correspondance établie par le code génétique[5] en une séquence, non plus de nucléotides, mais d'*acides aminés* pour constituer une *protéine*, ou parfois une version préliminaire inactive d'une protéine. La séquence d'ADN qui sert à la détermination de la séquence d'acides aminés de la protéine est appelée séquence *codante*. Les protéines sont les molécules actives de l'organisme, capables d'interactions pour réaliser des fonctions complexes qui peuvent conduire à la composition de multiples traits constitutifs du phénotype. Des technologies comme les puces à ADN (*microarray* en anglais) ou la spectrométrie de masse permettent d'observer de façon qualitative et quantitative les produits de la transcription, *i.e.* le *transcriptome*, et de la traduction, *i.e.* le *protéome*. A ce titre, transcriptome et protéome sont partie intégrante, au niveau moléculaire, du phénotype.

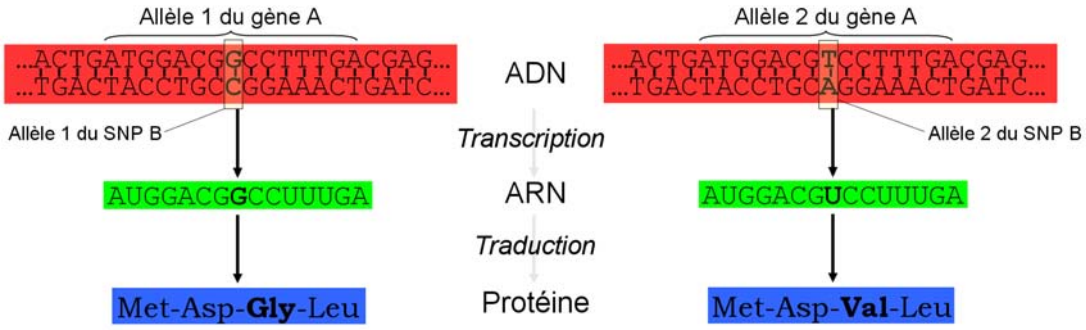

FIG. 1.1 – Représentation simplifiée des deux étapes de transcription et de traduction pour deux séquences d'ADN d'un gène (*i.e.* deux allèles) hypothétique A (à gauche l'allèle 1, à droite l'allèle 2) ne différant qu'en une seule position. En haut de la figure l'ADN est représentée sous sa forme native qui est celle d'un double brin dans lequel les nucléotides sont appariés selon les règles suivantes : A avec T et C avec G. On dit que les deux brins d'ADN ont des séquences complémentaires et on parle de paire de nucléotides à chaque position de la séquence. Les paires qui distinguent les deux allèles sur la figure sont G :C pour l'allèle 1 et T :A pour l'allèle 2. Au cours de la transcription, la copie de l'un des brins de l'ADN produit l'ARN_m dans lequel la différence entre les deux allèles est conservée. Enfin la traduction convertit l'ARN_m en une protéine dont les acides aminés sont enchaînés les uns aux autres en fonction de l'ordre des triplets sur la séquence de l'ARN_m et selon la correspondance donnée par le code génétique. La différence d'un nucléotide entre les deux ARN_m est ainsi lue comme une différence entre deux triplets GGC et GUC qui produit une différence entre les protéines traduites une différence d'acide aminé Gly (Glycine) en Val (Valine). Ainsi des génotypes différents portés par l'ADN sont exprimés grâce au double processus de transcription-traduction en deux protéines différentes qui pourront être responsables de deux phénotypes différents au niveau des fonctionnalités d'une cellule, d'un organe ou d'un organisme.

1.3 Les relations génotype–phénotype

L'étude des relations entre génotype et phénotype remonte aux expériences du moine Mendel en 1853. En croisant des souches de petits pois lisses ou ridés, il mit en évidence la transmission héréditaire

[5]Le code génétique, élucidé dans les années 60, met en correspondance de façon non ambiguë mais redondante les 64 triplets de nucléotides que l'on peut former à partir des 4 nucléotides constitutifs des ARN_m (A, C, G, U) et les 20 acides aminés constitutifs des protéines (`http://en.wikipedia.org/wiki/Genetic_code`). Les triplets (ou codons) sont ainsi lus et décodés par la machinerie cellulaire de biosynthèse des protéines qui enchaîne les uns aux autres les acides aminés correspondants selon l'ordre défini par la séquence de l'ARN_m. Le site de démarrage de la traduction sur une séquence d'ARN_m est le plus souvent déterminé par le triplet d'initiation AUG. La traduction s'arrête lorsque la machinerie cellulaire rencontre un triplet dit non-sens qui ne correspond à aucun acide aminé (UAA, UAG ou UGA).

de facteurs génétiques, *i.e.* le génotype, responsable de l'apparition de traits observables, *i.e.* le phénotype. Au début du XX$^{\text{ième}}$ siècle, le biologiste Morgan fonda la théorie chromosomique de l'hérédité en associant des altérations visibles des chromosomes géants d'une espèce de mouche (*Drosophila melanogaster*), à des modifications du phénotype (yeux blancs au lieu de rouges). Ensuite, les microbiologistes Griffith et Avery en 1928 démontrèrent en manipulant deux souches de pneumocoques que la molécule d'ADN était le support du matériel génétique. Ces trois découvertes novatrices de la génétique ont ouvert la voie à l'étude des relations génotype–phénotype qui vise à comprendre la part déterminée par le matériel génétique dans les traits qui composent un individu.

En médecine, l'étude des relations génotype–phénotype a donné lieu à l'exploration du domaine des maladies génétiques. Un exemple classique d'une telle maladie est la *drépanocytose*, causée par la modification, ou *mutation*, d'un seul nucléotide sur le gène de l'*hémoglobine*. Cette mutation observée chez les individus malades entraîne une anomalie dans la protéine hémoglobine alors responsable de la drépanocytose. Les conséquences au niveau du phénotype peuvent être observées à l'échelle microscopique à commencer par la structure anormale de la protéine, puis la déformation des globules rouges qu'elle induit. Les conséquences de celles-ci sont observées à une échelle macroscopique : c'est le cas d'occlusions des capillaires sanguins provoquées par la forme anormale (en faucille) des globules rouges, ou encore une résistance à la malaria également expliquée par cette forme originale.

Les technologies d'exploration des génomes, transcriptomes, et protéomes, permettent l'acquisition de nouvelles connaissances sur la séquence du génome et sur la réelle complexité des mécanismes de régulation de son expression. En effet, les relations génotype–phénotype sont plus compliquées qu'il n'a pu paraître dans un premier temps. Ainsi, il faut souvent que coexistent plusieurs facteurs génétiques pour expliquer un trait du phénotype [vHY04]. Parmi ces facteurs génétiques, certains ont des rôles modulateurs, parfois indirects, sur le phénotype. De plus, le phénotype associé à une maladie peut résulter de la co-occurence complexe de nombreux traits ou signes cliniques (c'est notamment le cas du *syndrome métabolique* [Mau06]). Dans ce cas, l'apparition de chaque signe clinique composant le phénotype peut être associée à de multiples facteurs d'origine génétique, chacun soumis à l'interaction d'autres facteurs génétiques, mais également à celle de facteurs environnementaux et comportementaux.

La caractérisation des relations génotype–phénotype constitue l'un des enjeux majeurs de la génomique. En effet son objectif ne se limite pas à l'étude du génome comme entité isolée, mais s'étend à l'élucidation des relations complexes qui existent entre la séquence et la structure du matériel génétique et le déploiement des fonctions des molécules biologiques dans la cellule et l'organisme.

1.4 Les sources de données et les vocabulaires contrôlés relatifs aux relations génotype–phénotype

Nous distinguons ici deux catégories de sources de données répertoriant des relations génotype–phénotype. Premièrement des sources constituées sur la base de publications scientifiques rapportant des relations génotype–phénotype, et deuxièmement des sources regroupant des jeux de données brutes qui ont permis de dériver de telles relations. Les deux sources évoquées ci-dessous illustrent respectivement ces deux catégories.

OMIM La base de données *OMIM*[6] (*Online Mendelian Inheritance in Man*) regroupe de nombreuses données sur les relations génotype–phénotype mises en évidence dans le cadre de l'étude des maladies génétiques. La plupart des entrées d'OMIM décrivent soit un gène, et détaillent alors son implication dans une ou plusieurs maladies, soit une maladie, et détaillent les rôles respectifs dans celle-ci de un ou plusieurs gènes. Le contenu de cette base de données est relativement peu structuré puisque ses entrées se

[6] `http://www.ncbi.nlm.nih.gov/omim/`

composent de textes courts, en langage naturel, répartis en différentes catégories (entre autres signes cliniques, mode de transmission, explication moléculaire, corrélations génotype–phénotype). OMIM s'appuie sur les publications scientifiques décrivant ces associations et résulte d'un travail de collecte initié dans les années 60, d'abord sous la forme d'un catalogue papier [McK98].

dbGaP Une source de données apparue plus récemment est *dbGaP*[7] (*database of Genotype and Phenotype*) dont l'objectif est le regroupement et le partage de jeux de données récoltés pour mettre en évidence des associations génotype–phénotype [MFJ+07].

Une limite actuelle de ces ressources est la faible structuration des données et notamment celle des termes utilisés pour décrire la notion complexe de phénotype. En effet la description d'un phénotype est construite sur des observations soumises à la subjectivité de l'observateur. L'un des objectifs de dbGaP est de réduire ce biais grâce à la mise à disposition des données brutes dont sont issues les descriptions des phénotypes. De façon complémentaire, une manière d'homogénéiser la description de phénotypes est de proposer un vocabulaire de référence (ou *vocabulaire contrôlé*) dont les termes pourront être utilisés et composés pour décrire de façon structurée un phénotype. Suivant cet objectif, différents groupes de recherche s'emploient à construire des vocabulaires plus ou moins consensuels pour permettre une description homogène des phénotypes. C'est par exemple le cas des vocabulaires contrôlés *PATO*[8], *Mammalian Phenotype*[9], ou *Plant Trait*[10].

[7] http://www.ncbi.nlm.nih.gov/sites/entrez?db=gap
[8] http://bioontology.org/wiki/index.php/PATO:Main_Page
[9] http://www.informatics.jax.org/searches/MP_form.shtml
[10] http://www.gramene.org/plant_ontology/

2 Les variations génomiques

Au sein d'une même espèce le génome présente de grandes similitudes, c'est pourquoi on parle par exemple du *génome humain* ou du génome de la *mouche à fruit* (Drosophila melanogaster). Cependant chaque être humain présente une version unique de ce génome humain[11]. Pour donner un ordre de grandeur, certains auteurs estiment à 99,9% le taux de nucléotides[12] similaires parmi les 3,2 milliards qui composent le génome humain, ce qui signifie que le 0,1% restant suffit à déterminer les différences entre les êtres humains [KN01].

2.1 Définitions

Les variations génomiques sont des régions du génome clairement localisées dont la composition en nucléotides est susceptible de varier entre les individus d'une même espèce.

La notion d'*allèle* correspond à la version d'un gène, et par extension à la version d'une variation génomique. Les organismes *diploïdes*, comme l'être humain, possèdent deux versions différentes du génome : une première héritée de la mère et une deuxième du père. Aussi un être humain est susceptible de porter deux versions différentes, *i.e.* deux allèles différents, de chaque gène. Si deux allèles distincts peuvent être portés par un même individu, de nombreuses autres versions peuvent être observées chez des individus distincts. Les parties gauche et droite de la Figure 1.1 montrent deux allèles d'un même gène dont la différence repose sur la substitution d'un seul nucléotide : un A à gauche et un C à droite[13].

La majeure partie des variations génomiques (plus de 90%) se limite à la variation d'un seul nucléotide entre deux allèles. Ce type particulier de variation est appelé *Single Nucleotide Polymorphism* en anglais ou *SNP* [KN01], *i.e.* au sens strict, un substitution d'un seul nucléotide dont la fréquence est supérieure à 1% dans la population dans laquelle il est étudié. Dans cette thèse, nous ne ferons pas cette distinction, très dépendante de l'échantillon des individus observés, et appellerons variation génomique ou variant l'ensemble des variations inter-individuelles du génome et SNP les variations ponctuelles sans prendre en considération leur fréquence. En revanche, nous éviterons le terme *mutation* hérité de l'étude des maladies génétiques et qui à ce titre correspond à une variation génomique associée à la survenue d'une maladie.

2.2 Les sources de données relatives aux variations génomiques

Les méthodes de séquençage et d'hybridation moléculaire permettent l'acquisition à haut débit de données relatives aux variations inter-individuelles d'un génome. Les données résultant de ce genre d'analyse du génome sont stockées et parfois partagées dans diverses bases de données dont le contenu se recouvre partiellement. Certaines de ces bases, relatives à l'être humain, sont présentées ci-après.

dbSNP La base de données *dbSNP*[14] du NCBI contient plus de 9 millions de variations génomiques humaines, et constitue la plus grande source de variations disponible sur le Web [SWK+01]. En plus de contenir les variations qui lui sont directement soumises, dbSNP intègre des données provenant d'autres

[11] Abstraction faite des clones et des vrais jumeaux.

[12] Pour être exact il s'agit de *paires de nucléotides* puisque l'ADN est composé d'un double brin de nucléotides complémentaires.

[13] Pour être exact il s'agit de la substitution des paires de nucléotides complémentaires : A-T et C-G. Par convention seul le nucléotide du brin *sens* est utilisé pour décrire l'allèle. Ce brin sens est celui dont la séquence est transcrite en ARN_m puis traduite pour donner la protéine.

[14] `http://www.ncbi.nlm.nih.gov/projects/SNP/`

grandes bases de données de variations génomiques comme les bases NCI CGAP-GAI[15], HGVBase[16], HapMap[17], Perlgen[18]. Une fois intégrées à dbSNP, certaines de ces bases sont amenées à disparaître. Un avantage stratégique de dbSNP est de faire partie intégrante des bases de données du NCBI (avec entre autres GenBank, PubMed, Gene, Human Genome Project Data) et à ce titre d'être interrogeable par le système fédéré *Entrez* [Bax06]. L'alimentation de dbSNP par des processus automatiques, le manque d'annotations manuelles des entrées rendent inégales la qualité et la validation des données qu'elle contient [MZCC04]. Il est important de noter que malgré son nom dbSNP ne répertorie pas seulement les SNP comme ils sont définis au sens strict *i.e.* la substitution d'un seul nucléotide dont la fréquence est supérieur à 1%. En effet, dbSNP répertorie les polysubstitutions, les insertions/délétions, et les variations plus complexes quelque soit leur fréquence d'observation dans les populations.

OMIM Comme décrit dans la section 1.4 de ce chapitre, OMIM contient des données relatives aux allèles de gènes impliqués dans des maladies génétiques. La description moléculaire des différences entre allèles est inégale selon les entrées. De façon encore assez rare, certains allèles répertoriés dans OMIM sont reliés à la variation génomique correspondante répertoriée par dbSNP.

Les bases de données locus spécifiques De nombreuses bases de données *locus spécifiques*, *i.e.* relatives uniquement aux variations d'un locus[19], se sont développées de façon indépendante. Celles-ci contiennent le plus souvent les résultats d'investigations d'un groupe de recherche biomédicale (voir d'un consortium) spécialisé dans l'étude d'un gène, d'une fonction biologique ou d'une maladie génétique. Les initiatives intitulées *HGMD*[20] (*Human Gene Mutation Database*) et *The Way Station*[21] tentent de fédérer et de rendre public le contenu de ces nombreuses bases spécialisées [GSC+08].

2.3 Hétérogénéité des données relatives aux variations génomiques

Une variation génomique est localisée sur une position précise d'une séquence génomique (*i.e.* d'ADN). Cependant, lorsqu'elle affecte une région transcrite, la variation est propagée sur la séquence transcrite (d'ARN), et si elle affecte une région codante, elle est propagée également dans la protéine (séquence d'acides aminés). Ceci est illustré dans la Figure 1.1. Les bases de données biologiques représentent indifféremment les variations sur l'ADN, l'ARN ou les protéines en fonction souvent du type de séquence sur lequel elles ont été observées. De fait, ces bases représentent aussi bien la variation originale que ses répercussions. En guise d'illustration, la substitution d'une guanine en une thymine peut être représentée par G/T dans une séquence d'ADN, GGC/GTC dans le codon concerné, g/u dans l'ARN correspondant, Gly/Val dans la protéine traduite. D'une façon similaire, les représentations de la position de la variation diffèrent d'une base de données à l'autre en fonction de la séquence de référence et de la version de cette séquence. Pour exemple, la substitution G/T est localisée à la position 11 087 877 sur la séquence génomique du chromosome 19 dont le numéro d'accession dans la base de données *RefSeq*[22] est NC_000019, à la position 2 489 679 dans la séquence du contig NT_011295, et à la position 565 dans la protéine NP_000518 (sur le second nucléotide du codon qui code pour le 565ième acide aminé). La même substitution peut également être localisée à la position 26 747 dans une séquence associée au gène *LDLR*, ou encore à la position 108 dans le onzième exon de ce gène.

[15] http://gai.nci.nih.gov/cgap-gai/
[16] http://www.hgvbaseg2p.org/index
[17] http://www.hapmap.org/
[18] http://genome.perlegen.com/
[19] Un locus est une région déterminée sur le génome pouvant contenir aucun, un, ou plusieurs gènes.
[20] http://www.hgmd.cf.ac.uk/
[21] http://www.centralmutations.org/
[22] http://www.ncbi.nlm.nih.gov/RefSeq/

En plus des multiples référentiels utilisés pour décrire les variations, s'ajoute l'utilisation d'identifiants (ou numéros d'accession) propres à chaque base de données. Ainsi la variation décrite précédemment est identifiée dans dbSNP comme le polymorphisme rs28942082. Une syntaxe générique est recommandée par la *Société pour l'étude des Variation du Génome Humain*[23] (HGVS pour *Human Genome Variation Society*) selon laquelle notre variation est décrite par l'expression suivante :

NC_000019.8g.11087877G>T,

où NC_000019.8 est le numéro d'accession unique dans RefSeq de la séquence utilisée pour positionner le variant, la lettre 'g' signifie que la séquence en question est génomique par opposition à 'p' utilisée pour les séquences protéiques, 11087877 correspond à la position dans la séquence de référence, et G>T décrit la variation de nucléotide observée [dDA00]. En pratique, l'utilisation de cette nomenclature est restreinte à certains auteurs qui l'utilisent pour décrire les variations de façon univoque dans le texte de leurs publications scientifiques. D'autres nomenclatures liées au contexte historique de l'observation des variations persistent à la fois dans la littérature et les bases de données. Par exemple, notre variant est présenté dans OMIM comme la variation *FH NAPLES* ou "LDLR Gly544Val", c'est à dire selon des descriptions associées aux circonstances de sa première observation.

Enfin, les bases de données privées ou les bases de données locus spécifiques utilisent encore d'autres notations dites *non-conventionnelles* qui viennent grossir le nombre de descriptions possibles pour une même variation. La Figure 1.2 illustre les nombreuses façons de désigner une variation génomique dans les bases de données publiques et privées.

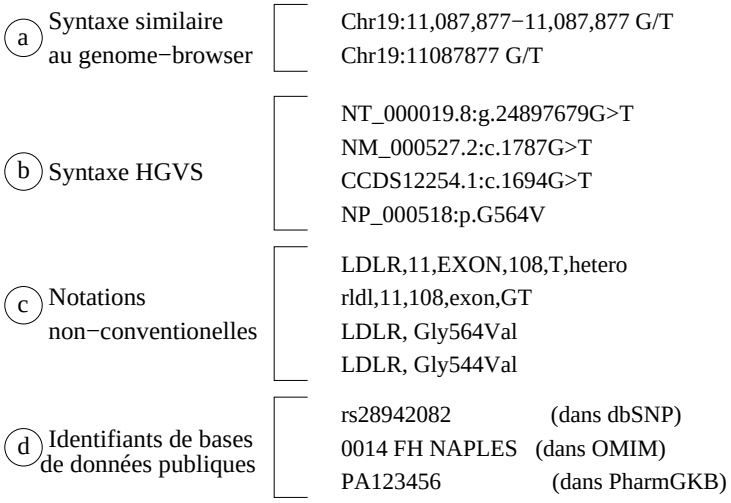

FIG. 1.2 – Diverses descriptions ou références pour une même variation génomique

L'une des raisons expliquant l'hétérogénéité de ces descriptions est leur origine : *(1)* certains projets de séquençage identifient de façon exhaustive les zones variables d'un génome ou de l'une de ses portions, *(2)* tandis que d'autres études plus ciblées identifient ponctuellement des *mutations génétiques*, *i.e.* des variations étroitement associées à la survenue d'une pathologie [Bar02]. L'identification de mutations génétiques a débuté préalablement à l'émergence des méthodes de séquençage, et a abouti à la constitution de nombreuses bases de données spécialisées et riches dont le spectre se limite aux mutations associées à un locus ou une maladie.

L'évaluation précise du recouvrement des contenus des bases de données de variations génomiques est crucial dans le cadre du développement de diagnostics génétiques et de l'exploration du *variome* (*i.e.*

[23] `http://www.hgvs.org/rec.html`

l'ensemble des variations du génome humain) [dDP03, RKC06, Spe08]. Cette tâche est rendue particulièrement délicate en raison du nombre important de descriptions différentes et pourtant équivalentes.

2.4 Les haplotypes

Un *haplotype* est un ensemble d'allèles de SNP (et éventuellement de gènes) voisins transmis conjointement à travers les générations. Les haplotypes sont des constructions statistiques établies sur une population donnée et obtenues par l'estimation des déséquilibres de liaison entre les allèles de SNP voisins. Bien qu'elles soient artificielles, ces constructions reflètent la réalité biologique selon laquelle le matériel génétique est transmis d'une génération à l'autre par blocs de séquences génomiques[Con05]. Ainsi, les variations génomiques présentes sur un même bloc présentent des valeurs qui sont liées les unes aux autres au fil des générations. En d'autres termes, on n'observe pas une distribution aléatoire des valeurs prises par les allèles au sein de ces blocs de séquences génomiques, mais au contraire un nombre fini de combinaisons de ces valeurs. Partant de ce principe, ces blocs sont reconstruits à partir de l'observation, dans une population, de groupes d'allèles associés pour des variations qui sont physiquement proches sur une séquence d'ADN. La Figure 1.3 illustre la notion d'haplotype et comment ils sont composés à partir des allèles présentés par des SNP voisins.

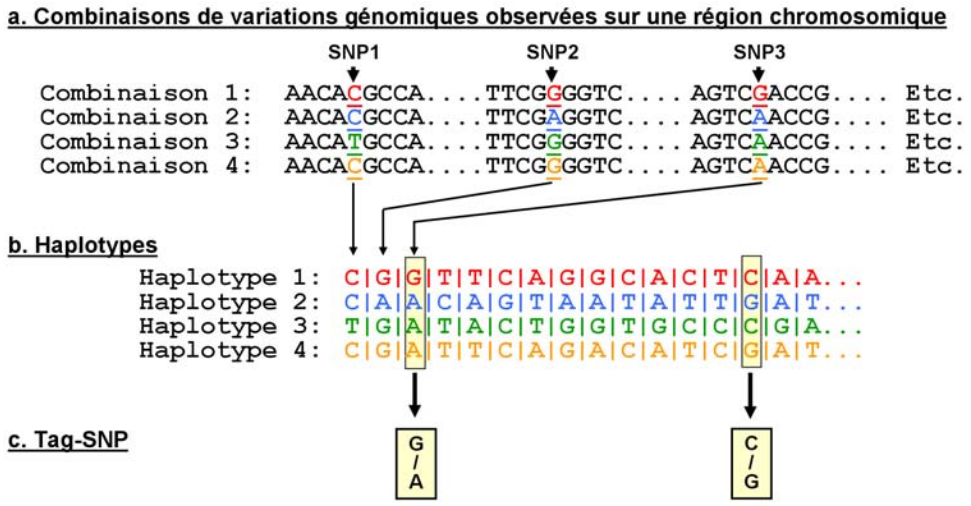

FIG. 1.3 – Haplotypes, tag-SNP et leur composition à partir des allèles de SNP voisins sur différentes versions d'un même chromosome. *Source :* http ://www.hapmap.org/

Le fait qu'un haplotype soit ainsi composé d'un ensemble associé d'allèles rend possible la distinction de certains allèles particuliers dont le génotypage suffit à déterminer les allèles présentés par le bloc de variations impliquées dans l'haplotype. Des outils statistiques permettent d'identifier ces SNP particuliers, appelés *tag-SNP* qui résument au mieux la composition d'un haplotype et de le distinguer des autres haplotypes observés sur un même bloc. Des exemples de tag-SNP sont représentés Figure 1.3.

HapMap est un projet de cartographie des haplotypes humains à partir du génotypage de variations génomiques dans 5 populations distinctes [Con03]. Les variations observées ainsi que leur fréquence d'observation sont disponibles dans la base de données associée au projet[24]. Ces données sont utilisées pour construire les haplotypes et identifier les tag-SNP à l'aide par exemple de l'outil *HaploView* [BFMD05].

[24] `http://www.hapmap.org/`

La notion d'haplotype est fréquemment utilisée pour réduire le nombre de variations génomiques à analyser dans des études (notamment sur les relations génotype–phénotype) qui s'intéressent aux variations de larges portions du génome. En effet, l'identification et le génotypage des seuls tag-SNP permettent de représenter les variations de blocs complets du génome et ainsi de réduire le nombre de variations génomiques à analyser. L'allèle présenté par chaque variation membre d'un haplotype peut par la suite être déduit à partir de l'allèle des tag-SNP et de la composition des haplotypes.

3 La pharmacogénomique

La réponse à un traitement médicamenteux est un phénotype particulier qui lui aussi est soumis à l'influence des facteurs génétiques. La *pharmacogénomique* s'attache à étudier ces facteurs génétiques particuliers et la façon avec laquelle ils influencent la réponse aux médicaments.

3.1 Définition

La pharmacogénomique est l'étude de l'ensemble des gènes ayant une influence sur la pharmacologie, elle s'intéresse notamment aux manifestations des variations génomiques de ces gènes, à l'interaction de ces variations dans la production d'un phénotype, et à l'influence d'un tel phénotype sur la réponse à un médicament [AK02]. Schématiquement, la pharmacogénomique peut être représentée comme l'étude des relations ternaires existant entre un *traitement médicamenteux*, un *génotype*, et un *phénotype* (Figure 4). Selon cette représentation, il est possible de considérer le traitement médicamenteux comme un facteur extérieur venant influencer la relation génotype–phénotype.

L'idée selon laquelle les gènes influencent la réponse aux médicaments date des années 50 durant lesquelles il fut observé que des réponses particulières aux médicaments pouvaient être transmises au sein d'une même famille ou étaient plus fréquentes au sein de certaines ethnies. Depuis, des études statistiques familiales et biochimiques ont renforcé cette hypothèse [ER04]. Cependant c'est seulement en 1988 que l'influence d'une variation dans la séquence d'ADN d'un gène sur le métabolisme d'un médicament a été mise en évidence [GSK$^+$88]. De nombreuses variations génomiques ont, par la suite, été isolées et associées à des effets différents d'un même médicament. La facilité grandissante à caractériser les variations génomiques inter-individuelles stimule l'investigation de la dimension génétique dans les essais cliniques des médicaments. Certains gènes impliqués dans les principales voies biologiques de transport ou d'élimination des médicaments sont plus particulièrement analysés.

Suivant cette évolution historique, l'étude initiale des caractères héréditaires associés à la pharmacologie fut appelée *pharmacogénétique*. L'émergence de la génomique a conduit à l'apparition du concept de pharmacogénomique, avec l'idée que la génomique offre la possibilité d'étudier l'origine et les conséquences des caractères héréditaires au niveau moléculaire.

Un exemple d'interaction pharmacogénomique décrite par Desmeules *et al.* [DGDM91] et Gasche *et al.* [GDF$^+$04] est l'influence des variations du gène *CYP2D6* dans la réponse à un traitement de codéine. La codéine est un opiacé prescrit, entre autres, pour son pouvoir analgésique. La codéine est physiologiquement métabolisée dans le foie en morphine, responsable de son effet analgésique. Il existe plusieurs versions fonctionnelles du gène *CYP2D6* dont les produits agissent différemment sur la transformation de codéine en morphine et permettent de distinguer plusieurs catégories d'individus (5) :

- les métaboliseurs lents, porteurs de variants à *activité faible* : par exemple Chr22 :40856638C>T et Chr22 :40854891G>A,
- les métaboliseurs rapides, porteurs de variants à *activité normale* ou *forte* : Chr22 :40853887C>T et les versions considérées normales des variations associées,
- les métaboliseurs ultra-rapides, porteurs de copies multiples de variants à *activité normale* ou *forte*).

Les métaboliseurs lents sont incapables de métaboliser efficacement la codéine en morphine, et en conséquence ne présentent pas l'effet analgésique attendu. Les métaboliseurs ultra-rapides, quant à eux, métabolisent la codéine avec une efficacité accrue qui entraîne une intoxication à la morphine.

En pharmacogénomique, le phénotype est également dépendant de la dose de médicament administrée. Ainsi suivant notre exemple, une dose plus élevée de codéine peut entraîner un effet analgésique chez les métaboliseurs lents et un effet toxique chez les métaboliseurs rapides. De nombreux exemples d'interactions de ce type peuvent être trouvés dans l'ouvrage *Pharmacogenetics* de Weber [Web97].

Certains des enjeux médicaux et industriels de la pharmacogénomique ont été abordés dans l'introduction de cette thèse. Les références suivantes [Flo05, NMG05, WMF+08] précisent ces enjeux et présentent les perspectives actuelles de la pharmacogénomique.

3.2 Les sources de données relatives à la pharmacogénomique

OMIM Les entrées de la base de données OMIM contiennent certaines données pharmacogénomiques. En effet, dans OMIM, les réactions adverses à des médicaments qui ont une origine génétique sont considérées au même titre que des maladies génétiques classiques.

PharmGKB PharmGKB[25] (*PharmacoGenomics Knowledge Base*) est la principale source de données publique pour la pharmacogénomique [HBWCH+08]. PharmGKB répertorie tout d'abord des données sur les relations entre médicament, phénotype, et gènes, données qui sont extraites manuellement de la littérature. De plus, PharmGKB contient des données sur les variations génomiques, les réseaux métaboliques impliqués dans ces relations, et des jeux de données réelles mêlant les données cliniques et génétiques de patients qui illustrent des éléments de connaissance pharmacogénomique. A ce titre PharmGKB peut être considérée comme une source de donnée particulière de variations génomiques et de relations génotype–phénotype. Une partie des variations génomiques répertoriées dans PharmGKB est reliée aux variations correspondantes dans dbSNP, mais un nombre également important de celles-ci sont soumises directement à PharmGKB et n'ont pas de correspondant dans les autres bases de données.

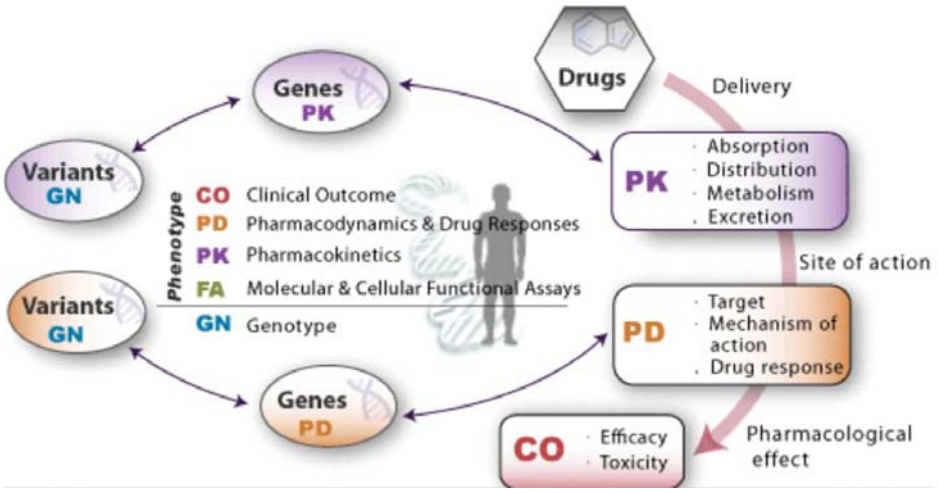

FIG. 1.4 – Représentation originale du schéma représentant les différentes catégories (CO, PD, PK, FA, GN) associées aux données de PharmGKB, et leurs principales associations (doubles flèches noires). *Source :* http ://www.pharmgkb.org/

Le schéma figurant sur la page d'accueil de PharmGKB et reproduit Figure 1.4 illustre assez bien la façon selon laquelle sont associées entre elles les données de PharmGKB et leur organisation en différentes catégories :

- CO : manifestations cliniques ou en anglais *Clinical Outcome*,
- PD : Pharmacodynamique et réponse au médicament, en anglais *Pharmacodynamics and Drug responses*,
- PK : Pharmacocinétique, en anglais *PharmacoKinetics*,

[25] `http://www.pharmgkb.org/`

– FA : Tests fonctionnels à l'échelle moléculaire et cellulaire, en anglais *molecular and cellular Functional Assays*,
– GN : Génotype, en anglais *Genotype*.

PharmGKB contenait en janvier 2008 des relations manuellement annotées entre plus de 600 gènes porteurs de variations, 450 maladies, et 500 médicaments [HBWCH+08]. L'amélioration continue des annotations relatives aux réseaux métaboliques, l'enrichissement de celles relatives aux variations génomiques, la mise en correspondance des variations génomiques répertoriées avec celles d'autres sources (dbSNP par exemple), et le développement de nouvelles campagnes d'investigations cliniques pour alimenter les jeux de données et l'état des connaissances [ORT08] font de PharmGKB une source qui fédère et stimule la recherche en pharmacogénomique.

4 Intérêt de l'utilisation de connaissances en pharmacogénomique

Les initiateurs de PharmGKB prévoyaient, à l'origine du projet, la constitution non pas d'une base de données relationnelle, mais d'une base de connaissances s'appuyant sur un langage de représentation des connaissances et associée à des mécanismes de raisonnement (comme nous le présentons chapitre 2, section 2.3) [ORS+02]. Face aux difficultés de mise en œuvre d'une telle approche, une architecture relationnelle plus classique a finalement été adoptée. Nous pensons qu'une approche à base de connaissances, comme celle initialement prévue pour PharmGKB, présente un intérêt particulier pour ce domaine.

L'état des connaissances en pharmacogénomique devrait bénéficier des données issues de l'exploration du génome. En effet des masses de données pertinentes pour ce domaine sont disponibles résultantes de l'étude des variations génomique, des relations génotype–phénotype, ou encore de la pharmacologie (voir par exemple les sources de données présentées dans les différentes sections de ce chapitre). Cependant l'interaction entre ces sous-domaines n'a pas forcement était considérée lors de leur exploration ou de la constitution des sources de données associées. Ainsi, il reste délicat d'analyser des résultats d'études pharmacogénomiques en prenant en considération simultanément les données associées à chacun de ces sous-domaines.

De plus, les méthodes d'analyses les plus utilisées dans le cadre de la pharmacogénomique demeure les méthodes statistiques classiquement utilisés pour les essais cliniques (les tests de corrélation, de régression, le déséquilibre de liaison par exemple [H.J02]). Ces derniers présentent des intérêts certains, mais ne permettent pas toujours d'apprécier ou d'explorer les larges volumes de données interconnectées tels que les bases de données biologiques ou les résultats d'une étude incluant le génotypage de l'ensemble du génome d'un panel de patient [YHTL08].

L'un des défis de la pharmacogénomique est justement de prendre en considération de larges volumes de données issues de différents sous-domaines spécialisés et interconnectés, pour leur associer un sens [AK02]. Gaines titre l'un de ses article par l'affirmation imagée selon laquelle *une once de connaissances vaut mieux que des tonnes de données* [Gai89]. Le travail présenté dans cette thèse s'inscrit dans cette idée et s'appuie sur l'hypothèse que la pharmacogénomique, et plus généralement la biologie moléculaire, peuvent tirer parti des méthodes de représentation des connaissances et d'extraction de connaissances. Un point commun à ces deux méthodes est, en effet, de permettre la découverte de connaissances implicites, voire nouvelles.

Chapitre 2

Etat de l'art

Ce chapitre présente, en section 1, le processus d'Extraction de Connaissances *à partir* de Bases de Données (ECBD) puis, en section 2, deux systèmes de représentation des connaissances en rapport avec les travaux menés dans cette thèse. Les sections 3 et 4 présente l'état de l'art des domaines concernés par les contributions de cette thèse : premièrement l'utilisation d'une représentation des connaissances, codée sous la forme d'une ontologie, pour guider l'intégration de données (section 3), secondement la notion d'Extraction de Connaissances *guidée par* les Connaissances du Domaine (ECCD).

1 Extraction de Connaissances *à partir* de Bases de Données – ECBD

1.1 Motivation et objectifs

L'Extraction de Connaissances à partir des Bases de Données (ECBD) est définie par Frawley *et al.* comme le processus non trivial d'identification de régularités (ou d'irrégularités) valides, nouvelles, potentiellement utiles, et porteuses de sens, au sein des données [FPSM91]. Concrètement il s'agit de l'utilisation de méthodes (souvent simplement d'algorithmes) de *fouille de données*, associées à une préparation des données préalables, et à une interprétation des résultats de fouille, afin d'extraire des connaissances pertinentes au regard des objectifs visés par l'analyste. Nous distinguons ainsi

(i) l'ensemble du processus d'ECBD qui inclut la préparation des données et l'interprétation des régularités extraites sous forme de connaissances, et

(ii) l'étape particulière de fouille de données dont le but unique et l'identification de régularités dans les données brutes.

La mise en œuvre de méthodes de fouille de données de façon "aveugle", *i.e.* sans étape de préparation appropriée des données ni d'interprétation experte des régularités extraites, est une utilisation dangereuse (comparée dans la littérature à une "pêche" ou une "drague") qui peut mener à l'extraction de régularités invalides, porteuses d'erreurs et ainsi à des interprétations inexactes.

L'ECBD est un processus comprenant plusieurs étapes, dont certaines impliquent une prise de décision de l'utilisateur, *i.e.* l'analyste qui conduit le processus. La Figure 2.1 présentée dans l'introduction de la thèse détaille le découpage classique du processus d'ECBD en plusieurs étapes. Du fait que la distinction et l'ordre des opérations de préparation de données peut fortement varier, nous proposons dans la Figure 2.1 une représentation simplifiée du processus centrée sur l'étape de fouille de données où nous distinguons une étape préalable globale de préparation des données et une étape finale d'interprétation. Le rôle de chacune de ces trois étapes ainsi que les opérations auxquelles elles font appel sont décrits dans les sections suivantes.

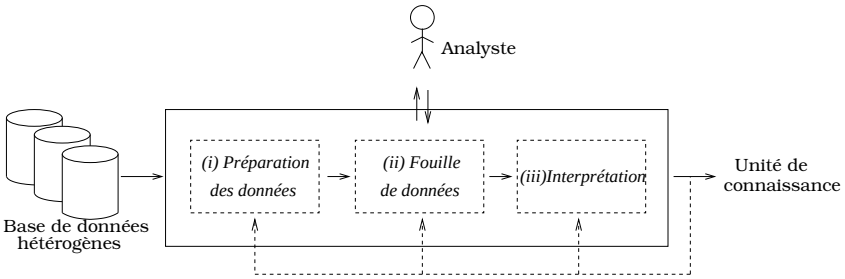

FIG. 2.1 – Représentation simplifiée du processus d'ECBD

1.2 Préparation des données

La préparation des données (ou *preprocessing* en anglais) est définie par l'ensemble des opérations qui permettent de convertir les données brutes en données préparées et adaptées à la méthode de fouille envisagée. L'intérêt principal de cette étape est d'améliorer la qualité des données (tout au moins en vue de la méthode de fouille choisie) et ainsi d'améliorer l'efficacité du processus d'ECBD . Les opérations de préparation peuvent être de différents types : l'*intégration des données*, le *nettoyage des données*, la *réduction des données*, la *transformation des données*. L'ordre de ces opérations varie souvent selon la stratégie d'ECBD adoptée. De la même façon, il n'est pas toujours évident de faire clairement la distinction entre les différentes opérations qui sont parfois entrelacées ou combinées. Par exemple, l'opération d'intégration de données nécessite souvent une étape préalable de nettoyage, la réduction des données peut consister en leur transformation en un format particulier, aussi le nettoyage peut conduire, au final, à une réduction de celles-ci.

Une bonne description de l'importance de ces étapes dans un processus d'ECBD est le chapitre de Brachman et Anand [BA96] du livre de Fayyad *et al.* [FPSSU96]. Un complément sur la mise en œuvre de ces opérations est le chapitre 3 du livre de Han et Kamber [HK01].

Les sections suivantes détaillent quatre types d'opérations relatives à la préparation des données.

1.2.1 Intégration de données

Un système d'intégration de données a pour rôle d'offrir à un utilisateur, ou à une machine, un accès *uniforme* et *transparent* à un ensemble hétérogène de données. L'intégration de données est alors le processus qui permet à un tel système l'accès homogène à un ensemble de données aux formats et aux localisations hétérogènes.

C'est une étape préliminaire nécessaire à la fouille de données. En effet, si les données à inclure dans l'analyse sont réparties dans des sources distinctes, il est nécessaire de les intégrer préalablement afin que l'algorithme de fouille puisse les prendre en compte simultanément.

L'intégration de données est d'autant plus intéressante que ses applications dépassent le cadre de l'ECBD. Ce processus est utilisé également dans le cadre de la recherche d'information, l'informatique décisionnelle, et l'étude des flux d'information (ou *workflow* en anglais) et trouve des applications dans de nombreux domaines où l'analyse des nombreuses données collectées présente un intérêt : la finance, les assurances, les systèmes de surveillance, le commerce, la médecine en sont des exemples. En bioinformatique l'intégration de données est une problématique de recherche active dont un des but est notamment de permettre l'utilisation conjointe des nombreuses sources de données biologiques qui ont vu le jour de façon indépendante et sans concertation [GS08].

La section 3 de ce chapitre propose un état de l'art sur les méthodes d'intégration de données et présente des solutions proposées dans le cadre de la bioinformatique. En effet la contribution présentée

chapitre 3 est précisément une proposition et l'application d'une méthode d'intégration opérationnelle, fondée sur des ontologies originales dans le domaine des variations génomiques et de la pharmacogénomique.

1.2.2 Nettoyage des données

En pratique, les données brutes sont souvent incomplètes, bruitées, voire incohérentes. L'opération de nettoyage a pour but de remplacer les *valeurs manquantes*, de filtrer le *bruit* (par exemple en éliminant les cas extrêmes) et de corriger les *incohérences* [HK01].

L'efficacité de certains algorithmes de fouille est très sensibles aux valeurs manquantes. Différentes approches peuvent être adoptées :

- ignorer les tuples dans lesquels des valeurs manquent. Cela peut s'avérer problématique lorsque le jeu de données initial est de petite taille.
- remplacer les valeurs manquantes par une valeur particulière, par exemple "Unknown", " ?". Cette méthode peut biaiser les résultats des algorithmes de fouille qui pourront considérer la valeur utilisée par défaut, disons "Unknown", comme représentative d'un concept intéressant.
- remplacer les valeurs manquantes par une valeur arbitraire. Ce peut être la moyenne des valeurs données à l'attribut dans le jeu de données, ou la moyenne d'autres attributs relatifs au tuple considéré, ou encore une valeur probable prédite par des méthodes d'inférence, de régression, d'induction sur la base d'autres données.

Les données brutes, et plus particulièrement celles mesurées expérimentalement, sont souvent accompagnées de bruit. Tout un ensemble de méthodes de filtrage et de lissage peut être mis en œuvre pour diminuer les effets de ce bruit.

Les incohérences dans les données peuvent être corrigées par des méthodes de comparaison avec les sources d'origine des données ou, si elles existent, par vérification des contraintes ou des dépendances connues entre données.

1.2.3 Réduction des données

La réduction de données vise à limiter la taille de la description des données en portant le moins possible atteinte à l'intégrité de l'information qu'elles contiennent. Diverses motivations peuvent amener à réduire les données :

- Certains algorithmes de fouilles de données produisent des résultats particulièrement volumineux et par conséquent compliqués et longs à interpréter. La réduction de données est une opération décisive dans un processus d'ECBD qui fait intervenir de tels algorithmes.
- D'autres algorithmes sont particulièrement gourmands en capacité de calcul et peuvent, en fonction de la taille du jeu de données, nécessiter des temps de calcul ou un espace mémoire incompatibles avec les conditions expérimentales (*i.e.* le temps et les machines disponibles).
- Certains jeux de données présentent un déséquilibre entre le nombre de tuples, relativement faible, et le nombre de valeurs distinctes, relativement élevé, que peuvent prendre les attributs associés. Il est possible d'imaginer le cas extrême où un jeu de données ne contient que des attributs à valeurs nominales et que chaque tuple présente une valeur différente pour chaque attribut. Dans ce cas particulier, les méthode de fouille ne pourront distinguer aucune régularité particulière sans l'utilisation d'une méthode extérieure. Des méthodes de réduction peuvent ici permettre de réduire la diversité entre les attributs qui caractérisent les tuples (en utilisant des valeurs plus générales qui seront partagées par plusieurs tuples par exemple).

Les stratégies de réduction de données incluent entre autres :

L'agrégation par cubes de données. Ce type de méthode souvent appliqué aux entrepôts de données, utilise des cubes de données qui permettent d'agréger des données multidimensionnelles dans le cadre d'analyses de type OLAP [AAD+96]. Par exemple, des données relatives aux ventes journalières d'une chaîne de grands magasins contenant des millions de transactions peuvent être agrégées en ventes mensuelles de certaines catégories spécifiques de produits.

La réduction de dimension. Ce type de réduction consiste à encoder les données dans un format plus compact entraînant, ou non, une perte d'information. Par exemple, l'*analyse en composante principale* est une méthode utilisée pour la réduction de dimension qui applique des projections des données initiales dans un espace de dimension inférieure.

La discrétisation. Il s'agit d'un ensemble de méthodes utilisées pour réduire le nombre de valeurs que peut prendre un attribut. Certaines méthodes automatiques de discrétisation s'appliquent aux attributs numériques et continus qu'elles partitionnent récursivement selon un échelonnage adapté au nombre et/ou à la répartition des valeurs. Ainsi l'éventail des valeurs que peut prendre un attribut comme la concentration d'une certaine substance pourra être divisé en plusieurs intervalles selon une construction d'histogramme. Certaines méthodes manipulant les histogrammes permettent par exemple de construire itérativement des histogrammes dotés d'intervalles de plus en plus importants permettant ainsi un ajustement de la discrétisation. Ces méthodes ne peuvent pas s'appliquer aux attributs discrets ou nominaux quand leur valeurs ne sont pas ordonnées (exemples : *couleur* = {rouge, vert, bleu} ou *allèle observé* = {AA, AT, AC, AG, TT, TC, TG, CC, CG, GG}). Dans ce cas il est cependant possible de construire manuellement un échelonnage ou une hiérarchie des attributs avec l'aide d'experts du domaines et/ou de méthodes heuristiques [HF94].

La sélection. La sélection de données a pour but d'identifier des sous-ensembles réduits de données sans en altérer la représentation originale. Il est possible de distinguer deux familles principales de méthodes de sélection de données [GE03, SIL05] :

- Les *méthodes de filtrage* qui la plupart du temps estiment un score d'intérêt pour les attributs du jeu de données qui permet de les classer et d'en supprimer les moins intéressants avant de les soumettre à la fouille. L'estimation du score peut être assurée à l'aide de méthodes heuristiques qui se fondent sur des mesures de significativité des attributs ou d'entropie comme par exemple le *gain d'information* [KJ97]. Les méthodes de filtrage les plus évoluées sont capables d'identifier les dépendances entre attributs et d'introduire cette composante dans le calcul du score d'intérêt (voir [YL04] pour un exemple). Le principal inconvénient de ces méthodes est qu'elles sont indépendantes de la méthode de fouille utilisée et ainsi qu'elles conduisent à estimer l'intérêt des attributs selon des critères différents de ceux utilisés par la méthode de fouille.
- Les *méthodes enveloppantes et intégrées* (*wrapper* et *embedded methods* en anglais) quant à elles sont dépendantes de la méthode de fouille considérée. De façon simplifiée, leur principe repose sur la constitution d'un ensemble fini de sous-ensembles de données qui seront chacun soumis à l'algorithme de fouille considéré. Alors le résultat de la fouille de chaque sous-ensemble de données est évalué et comparé aux autres afin de constituer de nouveaux sous-ensembles de données qui seront à leur tour testés lors d'une nouvelle itération. De façon non formelle, ces méthodes peuvent être considérées elles-mêmes comme des méthodes de fouille appliquées à des résultats partiels de la méthode de fouille considérée. Ces méthodes sont particulièrement coûteuses en calcul et le sont d'autant plus que le nombre d'attributs est élevé et que la méthode de fouille considérée demande elle-même des ressources importantes de calcul. Les algorithmes génétiques sont par exemple utilisés pour ce type de méthode de sélection de données [SIL05].

Dans le chapitre 4 nous proposons une approche de sélection dont la particularité est de tirer

bénéfice des connaissances du domaine disponibles.

Les connaissances de l'analyste peuvent aussi guider manuellement la sélection des données. La section 4 de l'état de l'art illustrera entre autres comment des connaissances formalisées peuvent être utilisées par l'analyste ou par des programmes en vue de la sélection des données. A ces travaux encore peu abondants s'ajoute la deuxième contribution de cette thèse qui consiste à proposer une approche de sélection des données guidée par les connaissances du domaine (chapitre 4 section 1).

1.2.4 Transformation des données

La transformation des données consiste en leur modification en une forme adaptée à la méthode de fouille envisagée.

Un premier exemple est la *normalisation* des données qui réside en leur échelonnage (*scaling* en anglais) sur différents intervalles ou ensembles de valeurs, comme de -1.0 à 1.0, de 0.0 à 1.0, ou {0, 1} ou encore {sous-exprimé, exprimé, sur-exprimé}.

Un second exemple de transformation est la *généralisation* qui, s'appuyant sur une hiérarchie de termes ou de concepts, permet de remplacer les valeurs d'attributs par leurs parents dans la hiérarchie, ce qui permet souvent de restreindre le nombre de valeurs possibles pour le nouvel attribut. Considérons par exemple un attribut "interaction avec un médicament" associée à une relation qui décrit des variations génomiques et peut prendre comme valeur les types de médicament avec lesquels la variation interagit. Les variations interagissant avec la codéine ou avec la morphine présentent la valeur "codéine" ou "morphine" pour cet attribut. Si ces deux exemple de valeurs sont remplacées par la valeur unique plus générale "opiacé" selon une hiérarchie de termes, les tuples (*i.e.* les variations) présentant la valeur "opiacé" pour cet attribut constitue un ensemble plus important que celles qui initialement avaient deux valeurs distinctes : "codéine" et "morphine". Cela peut permettre de réduire les différentes valeurs possibles pour certains attributs. Par contre, cette généralisation empêche alors de distinguer les variants qui interagissent avec la codéine de ceux qui interagissent avec la morphine.

L'*agrégation* est une transformation également intéressante lorsque les données peuvent être résumées ou agrégées pour être étudiées dans une dimension différente. Par exemple, le nombre de crises d'asthme d'un patient par semaine peut être agrégé pour être étudié au niveau mensuel ou annuel.

Le *lissage* qui revient à appliquer aux données une fonction d'approximation dans l'objectif d'éliminer les phénomènes locaux et de mettre en évidence les caractéristiques générales de celle-ci ou encore la *construction d'attributs* sont d'autres exemples de transformation de données [HK01].

1.3 Fouille de données

La fouille de données est l'étape de l'ECBD qui vise à extraire des régularités (ou des irrégularités) de l'ensemble de données préparées. Il existe de nombreuses méthodes de fouille différentes. Le choix de la méthode est déterminant et se fait essentiellement en fonction de l'objectif visé par l'analyste.

Les différents objectifs (ou *mining tasks* en anglais) de la fouille sont [HK01] :

- La description de classes (ou concepts) qui permet la caractérisation de classes ou la discrimination entre différentes classes.
- La recherche d'associations entre des attributs qui prennent des valeurs particulières de façon concomitante.
- La classification et la prédiction basées sur la définition d'un modèle à partir d'un jeu de données d'apprentissage.
- La construction de *clusters* qui regroupent les données en différents groupes selon des mesures de similarité.

– La détection de cas extrêmes révélant une forme d'irrégularité.

En pharmacogénomique, par exemple, les cliniciens sont intéressés par la découverte de facteurs permettant la discrimination d'un groupe de patients réagissant de façon adverse à un traitement par rapport à ceux pour qui aucune réaction néfaste n'est observée. Les biologistes, plus directement intéressés par l'étude du processus moléculaire des réactions pharmacogénomiques, peuvent être intéressés par la recherche d'associations entre, par exemple, un variant génétique, la réduction de l'activité d'une enzyme, et la concentration élevée d'une molécule dans le sang.

Les méthodes de fouille de données sont souvent classifiées en fonction des divers objectifs exposés ci-dessus. Il est également possible de distinguer les méthodes numériques des méthodes symboliques en fonction du type de données qu'elles manipulent. Cette distinction implique une différence dans les modalités de représentation, de manipulation, et de comparaison des données et des régularités résultantes.

– Les méthodes de fouille numériques comprennent, entre autres, les chaînes de Markov, les réseaux de neurones, les K-plus proches voisins, l'analyse en composante principale (ACP), les réseaux bayésiens, les algorithmes génétiques.
– Les méthodes de fouille symboliques comprennent, entre autres, l'extraction de motifs fréquents, la recherche de règles d'association, l'Analyse de Concepts Formels (ACF).

Une autre distinction est faite entre les méthodes dites supervisées et celles dites non-supervisées. Une méthode *supervisée* va proposer une *classification* des tuples/objets d'un jeu de données en s'appuyant sur un modèle préétabli à partir d'une base d'exemples ou d'échantillons de tuples/objets sélectionnés au hasard. Inversement, une méthode *non-supervisée* va produire un modèle sans apriori sur la seule information que lui apportent les tuples/objets. Dans ce cas la considération de nouveaux tuples/objets entraînera la mise à jour du modèle.

L'*apprentissage* est un domaine de recherche proche de la fouille de données utilisant des méthodes similaires mais avec une échelle et un objectif légèrement différents puisque les travaux d'apprentissage ne s'intéressent pas forcément aux larges volumes de données et que les résultats obtenus sont destinés plus particulièrement à la résolution de problèmes et à la prise de décision.

Les sections suivantes présentent trois méthodes de fouille de données symboliques qui extraient à partir de bases de données binaires soit un ensemble de concepts organisés en un *treillis* (*i.e.* un *ordre partiel*), soit des *motifs fréquents*, soit des *règles d'association*. Ces méthodes sont justement utilisées dans le chapitre 4 de cette thèse. La construction de treillis est présentée dans la section suivante (1.3.1) et les extractions de motifs et la recherche de règles sont décrites en la section 1.3.2. Enfin la recherche de règles d'association particulières dites Minimales Non-Redondantes est présentée section 1.3.3.

1.3.1 La classification par construction de treillis

Certaines méthodes de fouille de données s'apparentent à une classification et analyse des correspondances binaires entre une classe d'*objets* (ou individus) et une classe d'*attributs* (ou propriétés), informant ainsi, pour chaque paire objet-attribut, si l'attribut est observé pour l'objet ou non [GVM93, GW99]. Les attributs sont des propriétés qui qualifient les objets soit par leur présence ou leur absence, soit par une valeur qui a été discrétisée sous forme de plusieurs variables binaires. Ces variables sont regroupées dans des tableaux binaires (également appelés bases de données binaires ou contexte formel) qui décrivent les relations entre un ensemble d'objets et un ensemble d'attributs, où par exemple `(i,j) = 1` détermine que l'objet `i` présente l'attribut `j`. Cette relation est alors matérialisée par une croix "×" dans le tableau binaire correspondant.

L'Analyse de Concepts Formels (ACF) est une méthode d'analyse de données fondée sur les treillis de concepts (ou treillis de Galois) [GW99]. L'ACF a pour principe la transformation d'un *contexte formel*

$\mathcal{G}$ \ $\mathcal{M}$	A	B	C
1	×	×	
2		×	×
3	×		

TAB. 2.1 – Un premier exemple de contexte formel $\mathcal{K}$

en un ensemble de *concepts formels* organisés en un treillis. L'adjectif *formel* souligne ici le fait que les contextes et concepts sont manipulés en tant qu'entités mathématiques.

Pour définir la notion de treillis, il est nécessaire d'introduire au préalable les notions de *borne inférieure* et de *borne supérieure*.

Définition 2.1 *Soit (M,≤) un ordre partiel et A un sous-ensemble de M. Une* ***borne inférieure*** *de A est un élément s de M tel que* $s \leq a$, $\forall a \in A$. *Une* ***borne supérieure*** *de A peut être définie dualement. S'il existe un élément plus grand dans l'ensemble des bornes inférieures, celui-ci est l'****infimum*** *de A et noté inf A ou* $\wedge A$ *; dualement, une borne supérieure moindre est appelée* ***supremum*** *et est notée sup A ou* $\vee A$. *Si* $A = \{x, y\}$ *l'infimum inf A est également noté* $x \wedge y$ *et le supremum sup A est également noté* $x \vee y$.

Alors de façon générale, un treillis est un ordre $(\mathfrak{B}, \sqsubseteq)$, où la relation $\sqsubseteq$, appelée *relation de subsomption*, décrit un ordre partiel tel que chaque paire d'élément $\{x,y\}$ de $\mathfrak{B}$ possède une borne supérieure $x \vee y$ et une borne inférieure $x \wedge y$.

Définition 2.2 *Un ordre* $\underline{\mathfrak{B}} := (\mathfrak{B}, \leq)$ *est un* ***treillis****, si pour chaque paire d'éléments* $x - y$, *il existe toujours un infimum* $x \wedge y$ *et un supremum* $x \vee y$. $\underline{\mathfrak{B}}$ *est un treillis complet si son infimum* $\wedge X$ *et son supremum* $\vee X$ *existent pour chaque sous-ensemble X de* $\mathfrak{B}$. *Tout treillis complet* $\underline{\mathfrak{B}}$ *a un plus petit élément unique,* $\wedge\mathfrak{B}$, *et un plus grand élément unique* $\vee\mathfrak{B}$.

Dans le cadre de l'ACF un treillis est construit à partir d'un contexte formel défini comme suit :

Définition 2.3 (contexte formel) *Un contexte formel* $\mathcal{K}(\mathcal{G}, \mathcal{M}, \mathcal{I})$ *consiste en deux ensembles* $\mathcal{G}$ *et* $\mathcal{M}$ *et en une relation binaire* $\mathcal{I} \subseteq \mathcal{G} \times \mathcal{M}$ *entre* $\mathcal{G}$ *et* $\mathcal{M}$. $\mathcal{G}$ *est l'ensemble des* ***objets****, et* $\mathcal{M}$ *l'ensemble des* ***attributs*** *du contexte*[26]. $\mathcal{I}$ *est la* ***relation d'incidence*** *qui entre un objet g et un attribut m se note* $g\mathcal{I}m$ *ou* $(g, m) \in \mathcal{I}$.

Comme l'illustre le Tableau 2.1, un contexte formel est une base de données binaire qui peut être simplement représentée par un "tableau de croix", *i.e.* un tableau dans lequel les en-têtes de lignes correspondent aux noms d'objets, celles des colonnes aux noms d'attributs. La présence d'une croix au croisement de la ligne `i` et de la colonne `j` signifie que l'objet `i` présente l'attribut `j`.

La construction d'un treillis à partir d'un contexte formel se fonde sur la définition d'une fonction duale particulière qui permet d'associer à n'importe quel sous-ensemble d'objets un sous-ensemble d'attributs d'une part et d'autre part à n'importe quel sous-ensemble d'attributs un sous ensemble d'objets.

Définition 2.4 *Pour un sous-ensemble quelconque d'objets* $A \subseteq \mathcal{G}$, *nous définissons*

$$A' := \{m \in \mathcal{M} \mid \forall g \in A : (g, m) \in \mathcal{I}\} \tag{2.1}$$

[26]Plus précisément nous devrions dire "objets formels" et "attributs formels"

qui représente l'ensemble des attributs communs aux objets de A. Pour un sous-ensemble quelconque d'attributs $B \subseteq \mathcal{M}$, nous définissons de façon similaire

$$B' := \{g \in \mathcal{G} \mid \forall m \in B : (g, m) \in \mathcal{I}\} \tag{2.2}$$

qui représente l'ensemble des objets qui présentent tous les attributs de B.

La double utilisation de l'opérateur $'$, noté $''$ ($' : 2^{\mathcal{G}} \to 2^{\mathcal{M}}$ et $' : 2^{\mathcal{M}} \to 2^{\mathcal{G}}$) constitue la *connexion de Galois*. Il peut être montré que l'opérateur $'' : 2^{\mathcal{G}} \to 2^{\mathcal{G}}$, de même que $'' : 2^{\mathcal{M}} \to 2^{\mathcal{M}}$ sont des *opérateurs de fermeture* :

Définition 2.5 (opérateur de fermeture) *Soit X, X_1, et X_2 trois sous-ensembles de E (par exemple $\mathcal{G}$ ou $\mathcal{M}$). Un opérateur de fermeture h est une fonction (i) monotone croissante, i.e. $X_1 \subseteq X_2 \Rightarrow h(X_1) \subseteq h(X_2)$, (ii) extensive, i.e. $X \subseteq h(X)$, et (iii) idempotente, i.e. $h(X) = h[h(X)]$.*

*Alors un ensemble X de E est **fermé** si et seulement si $X = h(X)$*

Définition 2.6 (concept formel) *Un concept formel du contexte $\mathcal{K}(\mathcal{G}, \mathcal{M}, \mathcal{I})$ est une paire (A, B) avec $A \subseteq \mathcal{G}$, $B \subseteq \mathcal{M}$, et pour lequel la relation entre A et B est décrite par l'opérateur $'$ tel que :*

$$A' = B \text{ et } B' = A. \tag{2.3}$$

*A est appelé l'**extension** du concept (A, B), et B est appelée son **intension**. $\mathfrak{B}(\mathcal{G}, \mathcal{M}, \mathcal{I})$, appelé l'**ensemble des parties** de $\mathcal{K}$, est l'ensemble de tous les concepts formels du contexte $\mathcal{K}(\mathcal{G}, \mathcal{M}, \mathcal{I})$.*

Les propriétés particulières de l'opérateur de fermeture $''$ permettent de relier à chaque concept le concept fermé associé et permettent également de définir une relation d'ordre entre les concepts.

Définition 2.7 *Si (A_1, B_1) et (A_2, B_2) sont des concepts d'un contexte $\mathcal{K}(\mathcal{G}, \mathcal{M}, \mathcal{I})$, si $A_1 \subseteq A_2$ (et donc $B_2 \subseteq B_1$) alors (A_1, B_1) est appelé le **sous concept** de (A_2, B_2), et (A_2, B_2) le **super concept** de (A_1, B_1). Il en résulte la relation d'**ordre partiel** qui induit une hiérarchie entre ces deux concepts noté $\leq$*

$$(A_1, B_1) \leq (A_2, B_2) \tag{2.4}$$

L'ensemble des parties (i.e. de tous les concepts) $\mathfrak{B}(\mathcal{G}, \mathcal{M}, \mathcal{I})$ du contexte $\mathcal{K}$ organisé selon cet ordre et noté $\underline{\mathfrak{B}}(\mathcal{G}, \mathcal{M}, \mathcal{I})$ est le treillis de concept *(ou* treillis de Galois*) du contexte $\mathcal{K}$.*

Un treillis peut être représenté de différentes façons plus ou moins réduites. Un mode de représentation relativement riche inclut l'ensemble des concepts d'un contexte, *i.e.* chaque intension possible est déclinée pour former un concept. Cela permet la constitution du *treillis des parties* du contexte dont un exemple est représenté à gauche dans la Figure 2.2. Un mode plus classique et plus réduit consiste à ne représenter que les concepts fermés. Suivant l'exemple donné Figure 2.2 le concept ({2},{C}) présent dans le treillis des parties est éliminé et représenté par son fermé ({2},{A,C}) dans le treillis du centre de la figure. Un dernier mode, appelée *notation réduite* d'un treillis et définie dans [GW99] présente la particularité de ne signaler les objets que dans l'extension du concept le plus spécifique (*i.e.* le concept qui présente le plus d'attributs) dans lequel est inclus cet objet. Inversement, les attributs ne sont signalés que dans l'intension du concept le plus général (*i.e.* celui qui présente le moins d'attributs) dans lequel ils sont présents. Le treillis de droite de la Figure 2.2 est la notation réduite des deux premiers treillis.

La construction de treillis peut présenter différents avantages dans un processus d'ECBD [SWW98, Wil02, VMG04] :

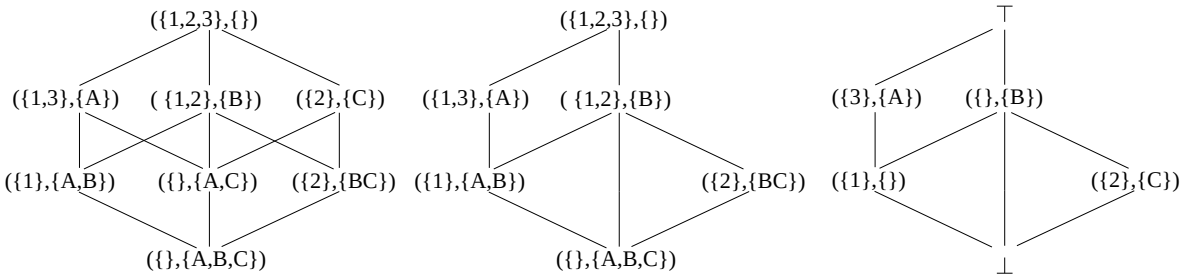

FIG. 2.2 – Différentes représentations du treillis associé au contexte $\mathcal{K}$ représenté dans le Tableau 2.1. De gauche à droite : le treillis des parties associé au contexte (où tous les sous-ensembles d'attributs sont représentés) ; treillis de Galois associé au même contexte ; treillis de Galois en notation réduite associé au même contexte.

– La structuration logique des données en concepts reflète la façon avec laquelle les humains conceptualisent un domaine. La proposition d'une hiérarchisation en concepts construite sans apriori à partir des seules données peut aider un analyste dans le cadre de l'extraction de connaissances.

– La formalisation des concepts peut permettre de proposer une traduction de la structure du treillis selon un formalisme logique, afin de pouvoir y appliquer des mécanismes automatiques de raisonnement.

– La construction du treillis peut servir d'étape préliminaire pour des algorithmes de fouille plus complexes. Ces algorithmes pourront alors tirer parti de l'organisation des concepts pour améliorer la rapidité de leur exécution, la gestion de la mémoire, ou les résultats produits.

– Les treillis sont également utilisés en recherche d'information (RI) [CR04, MDNST05]. L'utilisation de l'ACF en RI est, entre autres motivée par l'analogie évidente entre les associations objet/attribut de l'ACF et document/terme en RI. Selon cette analogie, les concepts formels peuvent être considérés comme des classes de documents qui correspondent à une requête de l'utilisateur. Alors les documents sont les objets caractérisés par des attributs qui sont les termes utilisés pour une requête. La relation de subsomption permet de guider le raffinement ou la généralisation de la requête (en y ajoutant/supprimant des termes) posée par un utilisateur en lui permettant de naviguer d'un concept à un autre.

– L'ACF est de plus en plus populaire en acquisition de connaissances à partir de textes. Le treillis peut constituer un complément aux méthodes de Traitement Automatique des Langues (TAL) en proposant une structure hiérarchique entre les concepts acquis par TAL. Les associations entre termes organisées en concepts dans un treillis peuvent permettre l'identification de nouveaux concepts ou d'instancier des concepts existants dans des processus de peuplement ou de construction d'ontologies [CHST04, BTN08].

REMARQUE : Nous distinguons dans cette thèse la notion de *concept formel*, entité mathématique résultant d'un processus d'ACF dont l'intension est une liste d'attributs, et les *concepts* utilisés en représentation de connaissances, notamment en Logique de Descriptions (LD). Eléments de base d'une ontologie, ces concepts ont pour intension une description formelle en LD qui définit les conditions d'appartenance à ce concept selon une certaine interprétation (voir section 2.2). Cependant une certaine analogie a pu conduire à des rapprochements entre ces deux notions et à des travaux situés à l'intersection des domaines de

$\mathcal{G}$ \ $\mathcal{M}$	A	B	C	D	E
1	×	×		×	×
2	×		×		
3	×	×	×		×
4		×	×		×
5	×	×	×		×

TAB. 2.2 – Un second exemple de contexte formel $\mathcal{K}$

l'ACF et des LD. De tels travaux [Rud06, BGSS07], détaillés au chapitre 4, sont à l'origine de la troisième contribution de cette thèse qui propose d'utiliser l'ACF pour découvrir de nouvelles connaissances au sein d'une base de connaissance formalisée en LD.

1.3.2 Motifs fréquents et règles d'association

En partant du même type de tableau binaire à partir duquel il est possible de construire un treillis, il est également possible d'extraire des motifs fréquents et de rechercher des règles d'association. Cette section présente rapidement ces deux méthodes.

L'extraction des motifs fréquents permet d'isoler, depuis un contexte formel, des ensembles d'attributs appelés *motifs* en accord avec un certain *support*. Ce support correspond au nombre d'objets qui partagent les attributs d'un motif et celui-ci doit être supérieur à un certain seuil, le *support minimum*, pour que le motif soit *fréquent*.

Sur la base des motifs fréquents, il est possible de construire des règles d'association, de forme générale $A \rightarrow B$ qui associe un sous-ensemble d'attributs A avec un second sous-ensemble d'attributs B. La règle peut alors être interprétée comme le fait que l'ensemble des objets avec les attributs de A présente également les attributs de B selon un certain support et une certaine *confiance* (défninie plus loin).

L'extraction de motifs fréquents

Définition 2.8 (motif fréquent) *Soit un contexte $\mathcal{K}(\mathcal{G}, \mathcal{M}, \mathcal{I})$ avec $\mathcal{G}$ un ensemble d'objets et $\mathcal{M}$ un ensemble d'attributs. Un* ***motif*** *est un ensemble d'attributs présenté par un objet. Il est dit que l'objet* ***contient*** *le motif. Le nombre d'attributs dans un motif détermine la* ***longueur*** *du motif. L'****image*** *du motif correspond à l'ensemble des objets qui contiennent le motif.*

Le ***support*** *d'un motif T est le nombre relatif d'objets qui contiennent ce motif parmi le nombre total d'objets $|\mathcal{G}|$ dans le contexte considéré $\mathcal{K}$, ainsi*

$$supp = \frac{|Image(T)|}{|\mathcal{G}|} \tag{2.5}$$

Le support peut être comparé à la probabilité $P(T)$ de trouver un objet contenant le motif T parmi l'ensemble des objets du contexte. Un motif est dit ***fréquent*** *si son support est supérieur ou égal à un seuil de fréquence arbitraire appelé* ***support minimum*** *(noté min_supp).*

Par exemple, si l'on considère le contexte formel représenté Tableau 2.2 et un $min\_supp = \frac{3}{5}$: {A} est un motif fréquent de longueur 1 et de support $\frac{4}{5}$; {AB} est de longueur 2, de support $\frac{3}{5}$, et fréquent ; {ABC} est de longueur 3, de support $\frac{2}{5}$, et non fréquent ; {ABCDE} est de longueur 5, de support 0, et non fréquent. On peut remarquer que le support diminue lorsque la longueur du motif augmente.

Si le nombre d'attributs de $\mathcal{M}$ est égal à n, le nombre de motifs possibles est 2^n (*i.e.* le nombre de sous-ensembles possibles à partir de $\mathcal{M}$). Aussi une extraction des motifs fréquents par le test systématique de la fréquence de chaque motif n'est pas envisageable. Cependant des algorithmes qui permettent de ne tester que certains sous-ensembles de motifs ont été développés et permettent d'extraire les motifs fréquents de grandes bases de données. L'algorithme `Apriori` est un outil classique d'extraction de motifs fréquents qui suit ce mode opératoire [AIS93]. `Apriori` s'appuie sur deux principes fondamentaux : *(i)* tout sous-motif d'un motif fréquent est un motif fréquent et *(ii)* tout super-motif d'un motif non fréquent est non fréquent. `Apriori` peut être résumé par ces deux opérations principales :

1. L'extraction des motifs fréquents commence par la recherche des motifs de longueur 1.

2. Les motifs fréquents sont enregistrés et combinés entre eux pour former des motifs *candidats* de longueur supérieure. les motifs non fréquents en 1. sont éliminés, et par conséquent aucun de leur super-motif n'est considéré. La fréquence des motifs candidats est testée pour constituer un nouvel ensemble de motifs fréquents et l'algorithme continue tant que de nouveaux candidats peuvent être formés.

L'algorithme 2.1 présenté plus loin dans ce chapitre en section 4.2 permet de suivre la succession des opérations de l'algorithme `Apriori` (la version présentée est enrichie par certaines opérations spécifiques à la problématique de cette section 4.2).

En guise d'exemple, nous pouvons réaliser pas à pas `Apriori` sur le contexte du Tableau 2.2 avec $min\_supp = \frac{3}{5}$. Les motifs fréquents de longueur 1 sont {A}($\frac{4}{5}$), {B}($\frac{4}{5}$), {C}($\frac{4}{5}$), {E}($\frac{4}{5}$). Le motif {D}($\frac{1}{5}$) n'est pas fréquent et est éliminé. Dans un second temps les motifs candidats de longueur 2 sont formés en combinant les motifs fréquents de longueur 1 : {AB}, {AC}, {AE}, {BC}, {BE}, ...puis leur fréquence est testée. Ainsi les motifs fréquents de longueurs 2 sont {AB}($\frac{3}{5}$), {AC}($\frac{3}{5}$), {AE}($\frac{3}{5}$), {BC}($\frac{3}{5}$), {BE}($\frac{4}{5}$), {CE}($\frac{3}{5}$). De la même façon les motifs candidats de longueur 3 sont formés puis testés pour donner les motifs fréquents de longueur 3 suivants : {ABE}($\frac{3}{5}$), {BCE}($\frac{3}{5}$). Enfin le seul motif candidat {ABCE} de longueur 4 est formé et testé mais son support ($\frac{2}{5}$) est inférieur à min_supp. Il est donc éliminé. Il n'y a plus de candidat, l'algorithme se termine.

Suivant un algorithme différent, les motifs fréquents peuvent facilement être extraits à partir d'un treillis. L'étape la plus contraignante est alors la construction du treillis, à partir duquel l'extraction des motifs fréquents est ensuite triviale. Elle correspond à un parcours en largeur dans le treillis, en partant du bas. La Figure 2.3 permet de distinguer facilement les motifs fréquents du contexte du Tableau 2.2 et de $min\_supp = \frac{3}{5}$.

La recherche de règles d'association

Définition 2.9 *Une règle d'association est de forme $T_1 \rightarrow T_2$, où T_1 et T_2 sont des motifs. T_1 est appelé la **prémisse** ou partie gauche de la règle, et T_2 est la **conclusion** ou partie droite de la règle. Le support de la règle $T_1 \rightarrow T_2$ est définie comme le support du motif $T_1 \cup T_2$, ainsi pour un contexte $\mathcal{K}(\mathcal{G}, \mathcal{M}, \mathcal{I})$*

$$supp(T_1 \rightarrow T_2) = \frac{|Image(T_1 \cup T_2)|}{|\mathcal{G}|} \tag{2.6}$$

La confiance *d'une règle $T_1 \rightarrow T_2$ est le rapport entre le support de la règle et le support de sa prémisse*

$$conf(T_1 \rightarrow T_2) = \frac{|Image(T_1 \cup T_2)|}{|Image(T_1)|} \tag{2.7}$$

La confiance peut être comparée à la probabilité conditionnelle $P(T_2|T_1)$, i.e. la probabilité de trouver parmi les objets du contexte qui contiennent le motif T_1, un objet contenant également le motif T_2.

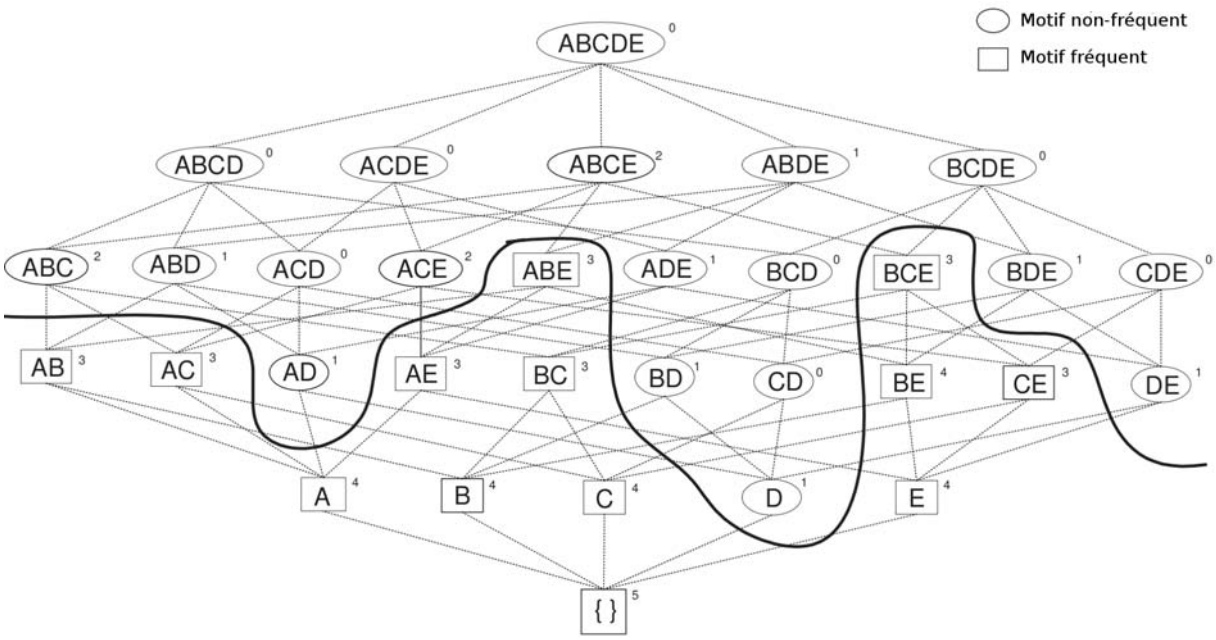

FIG. 2.3 – Treillis des parties associé au contexte $\mathcal{K}$ représenté Tableau 2.2. La ligne de séparation symbolise le support minimum ($min\_supp = \frac{3}{5}$) dissociant les motifs non fréquents, au dessus de la ligne, des motifs fréquents, en dessous. Le chiffre associé à chaque motif correspond au nombre d'occurences du motif dans $\mathcal{K}$. *Source* : exemple extrait de [Sza06].

Une règle est dite ***valide*** *si sa confiance est supérieure ou égale à un seuil de confiance arbitraire appelé* ***confiance minimum*** *(noté min_conf), et si son support est supérieur ou égal au support minimum (min_supp). Ainsi toute règle valide $T_1 \rightarrow T_2$ est basée sur un motif fréquent $T_1 \cup T_2$. Une règle est* ***exacte*** *si sa confiance est égale à 1, i.e. $supp(T_1 \cup T_2) = supp(T_1)$, sinon la règle est* ***approximative****. Les règles exactes sont également appelées des* ***implications****.*

Si l'on considère à nouveau le contexte du Tableau 2.2, avec $min\_supp = \frac{3}{5}$ et $min\_conf = \frac{3}{5}$, {AB} est fréquent, et la règle A→ B est valide ($supp = \frac{3}{5}$ et $conf = \frac{3}{4}$). La règle B→A est également valide ($supp = \frac{3}{5}$ et $conf = \frac{3}{4}$). Si l'on diminue le support de sorte que $min\_supp = \frac{2}{5}$ et $min\_conf = \frac{3}{5}$, le motif {ABCE} est fréquent, les règles AB→CE, CE→AB, AC→BE sont valides ($supp = \frac{2}{5}$ et $conf = \frac{2}{3}$ pour les trois), mais la règle BE→AC n'est pas valide ($supp = \frac{2}{5}$ et $conf = \frac{2}{4}$).

La construction des règles d'association valides depuis un motif fréquent (de longueur supérieure ou égale à deux) se fait de façon similaire à l'extraction de motifs fréquents. A partir d'un motif fréquent, la construction des règles débute par les règles dont la conclusion est de longueur 1, notées P \ {i}→{i} où {i} est un attribut seul et P \ {i} représente le motif P sans l'attribut {i}. Une fois ces règles construites, leur conclusions sont combinées pour donner de nouvelles règles candidates dont la conclusion est de longueur 2, notées P \ {ij}→{ij}. Ces nouvelles règles sont testées et le processus continue tant qu'il est possible de construire de nouvelles règles candidates.

Par exemple pour le contexte manipulé précédemment et $min\_supp = \frac{2}{5}$ et $min\_conf = \frac{2}{5}$, quand P = {AB}, les règles valides construites sont A→B ($min\_supp = \frac{3}{5}$; $min\_conf = \frac{3}{4}$) et B→A ($\frac{3}{5}$;$\frac{3}{4}$). Quand P = {ABC}($\frac{2}{5}$), les règles construites sont d'abord {AB}→{C}($\frac{2}{5}$;$\frac{2}{3}$), {AC}→{B}($\frac{2}{5}$;$\frac{2}{3}$), {BC}→{A}($\frac{2}{5}$;$\frac{2}{3}$), qui sont trois règles valides. Leurs conclusions peuvent donc être combinées pour produire les nouvelles conclusions {AB,AC,BC}, et les règles correspondantes {C}→{AB}($\frac{2}{5}$;$\frac{2}{4}$), {B}→{AC}($\frac{2}{5}$;$\frac{2}{4}$), {A}→{BC}($\frac{2}{5}$;$\frac{2}{4}$) qui sont également trois règles valides.

Le nombre de motifs et de règles générées est d'autant plus grand que le contexte permet d'associer un grand nombre d'objets et d'attributs. Cela rend délicate l'étape d'interprétation des unités extraites qui dans la plupart des cas est assurée par un analyste. Pour cette raison, il est crucial dans un processus d'ECBD et plus particulièrement lorsqu'il met en œuvre une extraction de motifs (ou une recherche de règle) de disposer de méthodes de filtrage des unités extraites. Dans ce but, de nombreux travaux se sont attachés à étudier les diverses mesures qui peuvent qualifier une règle [Fre98, LFZ99, TKS02, McG05]. En partant du fait que la confiance d'une règle A→B peut être considérée comme la probabilité conditionnelle P(B|A) (*i.e.* la probabilité de B sachant A) certaines de ces mesures peuvent être le fruit de calculs de probabilités comme par exemple l'*intérêt*, la *conviction*, ou la *dépendance* d'une règle. Une autre catégorie de mesures utilise des connaissances du domaine pour éliminer certaines règles [LHCM00, Sah02]. Ces méthodes sont alors dites *subjectives* par oppositions aux premières qualifiées d'*objectives*.

De la même façon que pour les motifs fréquents, la recherche de règles d'association, ainsi que le calcul de mesures, peuvent être facilement menés à partir d'un treillis de Galois.

La construction d'un treillis est une opération coûteuse en ressources informatiques et n'est pas nécessaire à l'extraction de motifs fréquents ou de règles valides pour lesquels des algorithmes plus efficaces existent. Cependant la structure mathématique qu'offre un treillis est intéressante pour caractériser des groupes particuliers de motifs et ainsi isoler différentes familles de motifs et de règles. Par exemple, le treillis proposé Figure 2.3 permet d'identifier de façon assez intuitive les règles exactes qui existent entre les motifs fréquents directement reliés et de même support. De cette façon les motifs {BCE} et {CE} directement reliés et de même support ($\frac{3}{5}$) traduisent l'existence de la règle exacte {CE}→{B}. La section suivante introduit une famille de règles particulières ainsi que la méthode qui permet d'en isoler les membres.

1.3.3 La famille des Règles Minimales Non-Redondantes

Cette section présente la famille particulière des règles d'association Minimales Non-Redondantes (notées $\mathcal{RMN}$) [Kry02, Sza06]. Le terme *famille* de règles vient du fait que nous distinguons cinq ensembles de règles parmi les Règles Minimales Non-Redondantes.

Briques nécessaires à la définition des $\mathcal{RMN}$

Pour pouvoir distinguer ces cinq ensembles particuliers de règles nous avons besoin de décrire des ensembles de motifs appelés *classes d'équivalence* et des motifs particuliers : les *motifs fermés fréquents* et les *générateurs fréquents*.

Définition 2.10 (classe d'équivalence) *Soit f une fonction qui associe à chaque motif $P \subseteq T$ l'ensemble de tous les objets qui contiennent le motif P : $f(P)=\{g \in \mathcal{G} \mid g$ contient $P\}$. Alors deux motifs $P, Q \subseteq T$ sont **équivalents** (noté $P \cong Q$) si et seulement si $f(P) = f(Q)$. L'ensemble des motifs équivalant à un motif P est appelé la **classe d'équivalence** de P et est notée*

$$[P] = \{Q \subseteq A \mid P \cong Q\} \tag{2.8}$$

Définition 2.11 (motif fermé fréquent) *La **fermeture** d'un motif X, notée $\alpha(X)$ est le plus grand super motif de X de même support que X.*

*Un motif X est alors un motif **fermé** si il n'existe pas de super motif Y de X (i.e. $X \subset Y$) de support identique à celui de X. Dans ce cas $X = \alpha(X)$. Les motifs fermés sont les motifs de longueur maximale au sein d'une classe d'équivalence parfois notée max$[P]$ pour une classe d'équivalence $[P]$.*

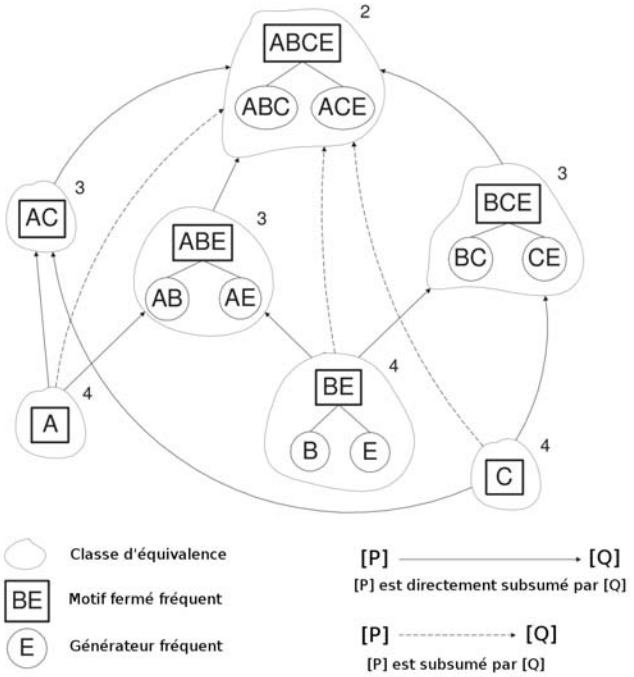

FIG. 2.4 – Classes d'équivalence, motifs fermés fréquents, et générateurs fréquents associés au contexte $\mathcal{K}$ représenté Tableau 2.2 ($min\_supp = \frac{2}{5}$). Les relations de subsomption entre classes d'équivalence sont déduites du treillis représenté Figure 2.3. *Source* : exemple extrait de [Sza06].

Un motif à la fois fermé et fréquent, suivant la Définition 2.8, est un ***motif fermé fréquent****.*

Définition 2.12 (générateur fréquent) *Un motif $P \in [P]$ est appelé* ***générateur*** *si P n'a pas de sous-motif dans $[P]$, i.e. si P n'a pas de sous-motif de support identique à P. En d'autres termes les générateurs sont les motifs de longueur minimale au sein d'une classe d'équivalence.*

Un ***générateur fréquent*** *est un générateur dont le support est supérieur ou égale à min_supp.*

Définition 2.13 (relation de subsomption entre classes d'équivalence) *Soit une classe d'équivalence $[P]$. La classe d'équivalence $[Q]$ est* ***ascendant*** *ou* ***subsumant*** *de $[P]$ si $max[P] \subset max[Q]$. La classe d'équivalence $[Q]$ est* ***ascendant direct*** *ou* ***subsumant direct*** *de $[P]$ si $[Q]$ est un ascendant de $[P]$ et qu'il n'existe aucune classe d'équivalence $[R]$ telle que $max[P] \subset max[R] \subset max[Q]$. La relation de subsomption sur les classes d'équivalence est transitive.*

La Figure 2.4 représente les classes d'équivalence, les motifs fermés fréquents, les générateurs fréquents, et les relations de subsomption entre classes pour le contexte représenté Tableau 2.2 et un support minimum de $\frac{2}{5}$. Dans cette figure, la classe d'équivalence dont le fermé est {C} est directement subsumée par la classe dont le fermé est {BCE} qui elle même est subsumée par la classe dont le fermé est {ABCE}. En revanche, il n'existe aucune relation de subsomption entre les classes d'équivalence dont les fermés sont {BCE} et {ABE}.

Les $\mathcal{RMN}$

Définition 2.14 (Base générique des règles exactes) *Soit FC l'ensemble des motifs fermés fréquents. Pour chaque motif fréquent $f \in FC$, FG_f est l'ensemble des générateurs fréquents de f. Nous définissons*

alors la base générique comme suit

$$\mathcal{BG} = \{r : g \rightarrow (f \backslash g) \mid f \in FC \wedge g \in FG_f \wedge f \neq g\}. \quad (2.9)$$

Définition 2.15 (Base informative des règles approximatives) *Soit FC l'ensemble des motifs fermés fréquents et FG l'ensemble des générateurs fréquents. Le motif noté $\alpha(g)$ représente le fermé de g. La base informative est alors*

$$\mathcal{BI} = \{r : g \rightarrow (f \backslash g) \mid f \in FC \wedge g \in FG \wedge \alpha(g) \subset f\}. \quad (2.10)$$

Définition 2.16 (Réduction transitive de la base informative) *Soit $\mathcal{BI}$ la base informative d'un ensemble de règles approximatives, et FC l'ensemble des motifs fermés fréquents. La réduction transitive de la base informative est*

$$\mathcal{BIR} = \{r : g \rightarrow (f \backslash g) \in \mathcal{BI} \mid \alpha(g) \textit{ est le sous-motif maximal de f dans } FC\}. \quad (2.11)$$

Définition 2.17 ($\mathcal{RMN}$) *L'ensemble des Règles Minimales Non-redondantes ($\mathcal{RMN}$) est défini comme*

$$\mathcal{RMN} = \mathcal{BG} \cup \mathcal{BI}. \quad (2.12)$$

Ainsi l'ensemble des $\mathcal{RMN}$ regroupe l'ensemble des règles exactes ($\mathcal{BG}$) et des règles approximatives ($\mathcal{BI}$).

Définition 2.18 ($\mathcal{RMNR}$) *L'ensemble des Règles Minimales Non-redondantes Réduites ($\mathcal{RMNR}$) correspond à la réduction transitive des $\mathcal{RMN}$:*

$$\mathcal{RMNR} = \mathcal{BG} \cup \mathcal{BIR}. \quad (2.13)$$

Les $\mathcal{RMN}$ constitue l'ensemble le plus grand de règles de cette famille, et $\mathcal{BG}$, $\mathcal{BI}$, $\mathcal{BIR}$, et $\mathcal{RMNR}$ en sont des sous-ensembles. Aussi il est facile, à partir des définitions précédentes de déduire les inclusions suivantes :

$$\begin{array}{ll} \mathcal{BIR} \subseteq \mathcal{BI} & \mathcal{RMNR} \subseteq \mathcal{RMN} \\ \mathcal{BG} \subseteq \mathcal{RMNR} & \mathcal{BI} \subseteq \mathcal{RMN} \\ \mathcal{BIR} \subseteq \mathcal{RMNR}. & \end{array}$$

La Figure 2.5 illustre la position relative des $\mathcal{RMN}$ et des $\mathcal{RMNR}$ par rapport à l'ensemble des règles d'association.

Calcul des $\mathcal{RMN}$

Nous pouvons remarquer que les définitions des $\mathcal{RMN}$ ne font intervenir que les deux ensembles de motifs particuliers : les motifs fermés fréquents et leur générateurs. De la même façon, les $\mathcal{RMN}$ peuvent être calculées à partir de ces deux seuls ensembles. L'algorithme *Zart* décrit par Szathmary *et al.* [Sza06, SNK07] permet d'isoler ces deux ensembles pour ensuite isoler les $\mathcal{RMN}$. Nous proposons en Annexe A un algorithme qui recherche les $\mathcal{RMN}$ et les $\mathcal{RMNR}$ à partir des motifs fermés fréquents et de leur générateurs.

Suivons un exemple à partir du contexte $\mathcal{K}$ (Tableau 2.2) avec $min\_supp = \frac{2}{5}$. La figure 2.4 permet de visualiser les motifs fermés fréquents et leurs générateurs dont nous allons nous servir pour cet exemple. Ainsi si nous considérons le générateur E de la Figure 2.4, deux types de règles peuvent être isolés. Un premier type correspond aux règles isolées au sein d'une classe d'équivalence et constitue la Base Générique ($\mathcal{BG}$), qui sont des règles exactes. En partant de E, la règle exacte E → B peut ainsi être isolée. Le second type de règles correspond aux règles isolées à partir des relations entre classes d'équivalence et

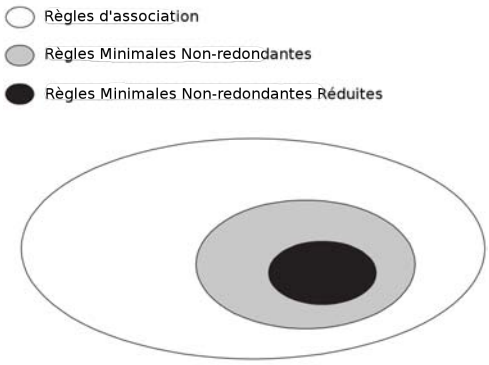

FIG. 2.5 – Représentation des inclusions successives de l'ensemble des Règles Minimales Non-redondantes Réduites ($\mathcal{RMNR}$) dans l'ensemble des Règles Minimales Non-redondantes ($\mathcal{RMN}$) puis de ce dernier ensemble dans celui de toutes les règles d'association.

constitue la Base Informative ($\mathcal{BI}$), qui sont des règles approximatives. Le générateur E permet d'isoler les règles E→ AB, E→ BC, et E→ ABC. Ensuite pour isoler les $\mathcal{RMNR}$, l'espace de recherche (des motifs fermés fréquents qui sont également super motifs du générateur considéré) est réduit aux classes d'équivalence qui sont reliées par une relation de subsomption *directe* (voir Définition 2.13), *i.e.* les relations de subsomption transitives ne sont plus considérées. De cette façon le générateur E ne permet d'isoler que trois règles : E→ B, E→ AB, et E→ BC. La règle E→ ABC isolée à partir d'une relation de subsomption indirecte n'est plus considérée. Aussi, si l'on retire les règles exacte des $\mathcal{RMNR}$ (E→ B selon notre exemple), nous obtenons la Base Informative Réduite ($\mathcal{BIR}$).

Intérêt des $\mathcal{RMN}$
Kryszkiewicz a démontré que les $\mathcal{RMN}$ et les $\mathcal{RMNR}$ constituent des représentations de l'ensemble des règles d'association qui sont *sans perte* (*i.e.* elles permettent de dériver la totalité des règles valides), *consistantes* (*i.e.* elles empêchent de dériver des règles non valides), et *informatives* (*i.e.* elles permettent de déterminer les paramètres des règles comme leur support et leur confiance). L'avantage principal des $\mathcal{RMNR}$ est de constituer l'ensemble le plus *concis* des règles d'association qui peuvent être extraites d'un contexte formel *sans perte* d'information.

C'est pour cette raison que nous utilisons la recherche des $\mathcal{RMNR}$ à partir d'un treillis comme méthode de fouille dans un processus d'Extraction de Connaissances *à partir* d'une Base de Connaissances présenté au chapitre 4, section 2.3.

1.4 Interprétation en unités de connaissances

L'étape d'interprétation du processus d'ECBD est également appelée le *post processing* en anglais. Elle consiste en la prise en charge des résultats bruts de la fouille de données, les unités extraites, en leur transformation pour leur interprétation et validation par l'analyste en unités de connaissance.

Cette étape est particulièrement limitante dans le processus d'ECBD car elle demande une implication importante de l'analyste qui doit interpréter des résultats de fouille potentiellement volumineux. La forme des unités extraites est différente selon la méthode de fouille utilisée : motif fréquent, concept formel, règle d'association, cluster par exemple. D'un point de vue pratique l'étape d'interprétation dépend fortement de la méthode de fouille utilisée puisque la forme des unités extraites dépend de celle-ci. Afin de faciliter l'interprétation, les résultats sont transformés pour faire l'objet d'une visualisation

graphique par exemple sous la forme d'un arbre de décision, d'une hiérarchie de clusters, d'un réseau de neurones, d'un treillis de concepts.

Une même forme d'unités extraites peut être utilisée pour étudier différents types de régularités. C'est l'*objectif* de la fouille qui dans ce cas oriente la lecture des unités extraites (*i.e.* du modèle) par l'analyste. La caractérisation, la discrimination, la recherche d'association, la classification, le clustering ou la détection des cas extrêmes sont les objectifs les plus souvent visés. Alors suivant l'objectif choisi, l'analyste s'intéresse à l'une ou l'autre des régularités observables à partir des unités extraites. Par exemple, les unités extraites résultant d'un clustering des K-plus proches voisins peuvent, selon l'objectif, être utilisées pour caractériser des groupes de données spécifiques ou pour décrire des associations entre données.

Au delà de l'objectif de la fouille, les connaissances attendues par l'analyste orientent l'interprétation. L'analyste peut alors être amené à filtrer parmi les unités extraites celles qu'il juge triviales, redondantes, dénuées d'intérêt, fausses en comparaison de ce qu'il souhaite trouver. Par exemple dans le cadre d'une recherche de règles d'association, un analyste souhaite découvrir des associations entre un phénotype, un génotype, et un traitement médicamenteux. Il peut éliminer les règles qui ne contiennent pas à la fois un critère correspondant à la classe <donnée du phénotype> (préalablement définie), un critère de la classe <donnée du génotype>, et un critère de la classe <traitement>. Ce genre de filtrage sur les résultats de fouille peut être assuré par un système qui tire parti de connaissances du domaine, pour permettre par exemple de distinguer les données qui relèvent du phénotype, de celles qui relèvent du génotype, ou d'un traitement.

1.5 Réutilisation des unités extraites

Les unités extraites finalement validées par l'analyste sont considérées comme unités de connaissance. Selon le processus d'ECBD initialement décrit par Frawley *et al.* [FPSM91], puis repris par Fayyad *et al.* [FPSS96], l'identification d'une unité de connaissance constitue un aboutissement du processus et est rarement réutilisée. En revanche, les unités extraites sont classiquement réutilisées lors des itérations successives du processus.

Le travail décrit dans cette thèse s'inscrit dans l'idée que les unités de connaissances doivent être formalisées dans un langage de représentation des connaissances et enregistrées dans une Base de Connaissances (BC) de sorte à pouvoir être réutilisées tout d'abord lors des itérations suivantes du processus, et ensuite dans le cadre d'autres applications qui peuvent tirer parti de connaissances formalisées. Nous présentons dans la section suivante 2 quelques notions de représentation des connaissances.

2 Représentation des connaissances et ontologies

Nous avons présenté dans l'introduction la distinction entre *données*, *information*, et *connaissance*. Ainsi les *bases de données*, étudiées depuis plusieurs décennies en informatique, permettent de structurer et de stocker des données brutes qui peuvent, dans le domaine de la génomique par exemple, être les résultats d'un séquençage automatique d'ADN, d'une analyse sur puce du niveau d'expression des gènes d'un tissu, ou encore d'une analyse par spectrométrie de masse du contenu protéique d'un échantillon du même tissu. Une *Base de Connaissances* (BC) est capable de stocker des données mais est également capable de leur associer une représentation formelle, *i.e.* associée à une sémantique clairement définie et conçue pour être interprétée par des programmes. Les connaissances peuvent ainsi spécifier des relations et des contraintes sur les données de telle sorte que les programmes puissent raisonner sur ces données pour en déduire de nouvelles connaissances. Les bases de connaissances s'appuyent sur des *langages de représentation des connaissances* afin non seulement de fournir une structure appropriée pour stocker les données mais surtout pour leur associer une interprétation du domaine considéré.

Cette section présente d'abord deux familles de langages de représentation des connaissances qui sont les *Représentations des Connaissances par Objet* (RCO) et les *Logiques de Descriptions* (LD) avant de définir les notions d'*ontologie* et de *Base de Connaissances*.

2.1 La Représentation des Connaissances par Objets

Le succès des Langages de Programmation à Objets (LPO) en informatique est souvent expliqué par les bénéfices qu'ils offrent en matière de génie logiciel grâce entre autres à la modularité, l'extensibilité ou la réutilisabilité des "objets informatiques". Cependant ce succès est certainement également dû à leur capacité naturelle à représenter les "objets du monde réel" [NED00]. Cette capacité n'a pas seulement favorisé l'adoption des LPO mais a également contribué au développement d'une famille de Représentation (ou de systèmes de représentation) des Connaissances par Objets (ou RCO) comme YAFOOL [DQ86], SHIRKA [ER95], et AROM [PGC+01]. Le coté intuitif de ces langages de RCO a notamment été utilisé pour permettre la représentation et la manipulation d'entités biologiques complexes dans [MVB+95] et [CCQF05] qui mettent respectivement en œuvre SHIRKA et AROM.

Objet, classe, attribut, facette et association Dans un formalisme de RCO l'élément de base est l'*objet*. Une *classe* permet de regrouper un ensemble d'objets ayant des propriétés communes appelés *attributs*. Les classes associent des *facettes* aux attributs pour les spécifier. Les facettes permettent *(1)* le *typage* des attributs, *i.e.* la précision d'un type ou d'un domaine de valeurs possibles pour un attribut ; *(2)* l'*inférence de valeur* pour un attribut *i.e.* l'association à des mécanismes capables de rattacher à une valeur à un attribut selon certaines contraintes ou calculs définis.

Les relations entre objets sont décrites par des *associations* qui peuvent être de deux types. Premièrement, les *attributs-liens* pour les relations binaires qui prennent la forme d'un attribut spécifique dont la valeur sera l'instance d'une classe. Deuxièmement, la *réification* d'association pour les relations *n*-aires, qui revient à considérer une association comme une classe dont les attributs sont les liens entre objets ou des attributs spécifiques qui qualifient l'association.

Spécialisation, partie-tout, et instanciation Les classes définies selon un RCO sont organisées selon une hiérarchie fondée sur une relation de spécialisation (apparentée à la subsomption décrite dans la section suivante). Une classe *descendante* d'une autre dans cette hiérarchie possède tous ses attributs (on parle alors d'héritage) et peut également présenter des attributs supplémentaires qui lui sont propres.

Les classes peuvent également être reliées selon des relations de composition ou d'agrégation par la relation *partie-tout* pour représenter le fait qu'un objet puisse être composé d'autres objets.

Les classes des RCO présentent la propriété de pouvoir être instanciées par un objet. L'objet en question devra alors présenter des valeurs pour les attributs définis dans la classe. Si au moins un attribut n'est pas valué alors l'instanciation est dite *incomplète*. L'ensemble des objets qui instancient une classe est appelée l'*extension* de cette classe.

Mécanismes de raisonnement Comme tout langage de représentation des connaissances, les RCO ont pour principal objectif de permettre le raisonnement sur les connaissances. Divers mécanismes de raisonnement sont associés aux langages de RCO :

- la *vérification de cohérence* qui teste les relations de spécification entre classes et d'instanciation entre classe et objet,
- la *classification d'instances* qui permet de trouver les classes auxquelles une instance **peut** appartenir,
- la *classification de classes* qui trouve les classes dont une classe particulière **peut** être la spécialisation,
- le *filtrage* qui recherche l'ensemble des objets satisfaisant des caractéristiques définies dans un filtre,
- le *raisonnement par classification* qui positionne une entité (une classe ou un objet) dans une hiérarchie de classes.

Les systèmes de RCO présentent l'avantage de proposer des mécanismes de raisonnement intéressants et de permettre une conceptualisation intuitive des entités considérées. Cette conceptualisation peut également facilement être représentée, voir automatiquement traduite, dans des représentations schématiques particulièrement lisibles comme le langage de modélisation UML [RBJ00]. Leur inconvénient est de ne pas présenter de véritable assise logique et de ne disposer que d'une expressivité relativement limitée, notamment comparé aux Logiques de Descriptions (LD) présentées dans la section suivante.

2.2 Les Logiques de Descriptions

Les Logiques de Descriptions (LD) constituent une famille de langages de représentation des connaissances fondée sur un formalisme logique. Les langages de LD sont des héritiers du système KL-ONE décrit en 1985 notamment pour surmonter les ambiguïtés sémantiques que présentaient les systèmes de représentations des connaissances préexistants (*i.e.* les réseaux sémantiques et les systèmes à base de frame) [BS85].

Comme les autres langages de représentation de connaissances, les LD sont utilisées pour représenter la conceptualisation d'un domaine d'application de façon structurée et en suivant une certaine sémantique. Leur avantage est premièrement que cette sémantique est clairement définie et deuxièmement qu'elles disposent de constructeurs logiques variés assurant une expressivité relativement riche (par exemple par rapport à la plupart des langages de RCO).

Les différents membres de la famille des LD se distinguent les uns des autres notamment par la liste des constructeurs qu'ils proposent. Le Tableau 2.3 liste les constructeurs de base communs à la plupart des LD. Les constructeurs sont associés à des symboles ($\mathcal{AL}$, $\mathcal{U}$, $\mathcal{C}$, ...) qui sont assemblés pour former les noms des LD qui les contiennent. De cette façon une logique de descriptions de base, appelée $\mathcal{AL}$ n'inclut pas l'union de concepts comme constructeur (associé au symbole $\mathcal{U}$), mais la logique qui contient les constructeurs inclus dans $\mathcal{AL}$ associé au constructeur permettant l'union de concept existe également et s'appelle $\mathcal{ALU}$. Le lecteur pourra trouver une note complète sur les conventions de nommage des LD dans les annexes de [BCM+03] (page 504).

***TBox* et *ABox* : concept, rôle, individu, et axiome** Une Base de Connaissances (BC) en LD est composée de deux éléments : la *TBox* et la *ABox*. Le Tableau 2.4 est un exemple de BC exprimée en LD.

Nom du constructeur	Syntaxe	Sémantique	Symbole
Concept universel	$\top$	$\Delta^{\mathcal{I}}$	$\mathcal{AL}$
Bottom	$\bot$	$\varnothing$	$\mathcal{AL}$
Intersection	$\mathtt{C} \sqcap \mathtt{D}$	$\mathtt{C}^{\mathcal{I}} \cap \mathtt{D}^{\mathcal{I}}$	$\mathcal{AL}$
Union	$\mathtt{C} \sqcup \mathtt{D}$	$\mathtt{C}^{\mathcal{I}} \cup \mathtt{D}^{\mathcal{I}}$	$\mathcal{U}$
Négation	$\neg\mathtt{C}$	$\Delta^{\mathcal{I}} \setminus \mathtt{C}^{\mathcal{I}}$	$\mathcal{C}$
Restriction universelle	$\forall \mathtt{R}.\mathtt{C}$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y.\ (x,y) \in \mathtt{R}^{\mathcal{I}} \rightarrow y \in \mathtt{C}^{\mathcal{I}}$	$\mathcal{AL}$
Restriction existentielle	$\exists \mathtt{R}.\mathtt{C}$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y.\ (x,y) \in \mathtt{R}^{\mathcal{I}}$	$\mathcal{E}$

TAB. 2.3 – Syntaxe et sémantique associées aux constructeurs de concepts les plus simples en LD . Les constructeurs disponibles dans la logique de base $\mathcal{AL}$ n'ont pas de symbole propre, pour les autres le symbole correspondant est donné dans la quatrième colonne. L'annexe B décrit une liste plus complète des constructeurs de concepts ainsi que de certains constructeurs de rôles.

(Ax1) `Personne` $\sqsubseteq \top$
(Ax2) `TraitementMédicamenteux` $\sqsubseteq \top$
(Ax3) `Patient` $\sqsubseteq$ `Personne`
(Ax4) `PatientSousTraitement` $\equiv$ `Patient` $\sqcap \exists$ `aPourTraitement` .`TraitementMédicamenteux`
TBox

(Ax5) `Patient(adrien)`
(Ax6) `TraitementMédicamenteux(cureDAntibiotique)`
(Ax7) `aPourTraitement(adrien, cureDAntibiotique)`
ABox

TAB. 2.4 – Un exemple de Base de Connaissances écrite en LD

La *TBox* constitue une *terminologie*, *i.e.* le vocabulaire d'un domaine d'application. Ce vocabulaire est constitué *(i)* de *concepts* qui correspondent à un ensemble d'*individus* et peuvent être comparés aux prédicats unaires des logiques des prédicats ; et *(ii)* de *rôles* qui représentent des relations binaires entre les individus et peuvent être comparés à des prédicats binaires. Une particularité des LD notamment par rapport aux langages de RCO est que deux types de concepts et de rôles sont distingués : les concepts et rôles *atomiques*, et les concepts et rôles *définis* :

- les concepts et rôles atomiques sont décrits seulement par leur nom, comme par exemple le concept `Personne` et le rôle `estTraité` dans la BC représentée Tableau 2.4,
- les concepts et rôles définis sont décrits par leur nom auquel est associé une description complexe. Dans la BC proposée en exemple, le concept `PatientSousTraitement` est le seul concept défini. Le langage avec lequel sont décrits les concepts et rôles est la LD choisie pour cette BC. Ces descriptions complexes sont appelées les *axiomes terminologiques*.

La sémantique associée aux concepts est définie par le biais d'une *interprétation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. Le domaine d'interprétation $\Delta^{\mathcal{I}}$ de $\mathcal{I}$ est un ensemble non vide, et la fonction d'interprétation $\cdot^{\mathcal{I}}$ associe à chaque concept atomique $\mathtt{A}$ un ensemble $\mathtt{A}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ et à chaque rôle atomique $\mathtt{R}$ une relation binaire $\mathtt{R}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. L'extension de la fonction d'interprétation aux concepts (et rôles) définis est déduite de façon inductive par la sémantique associée aux constructeurs de concepts (et de rôles) présentés Tableau 2.3.

Type d'axiome	Syntaxe	Sémantique
Définition de concept	$\mathtt{C} \equiv \mathtt{D}$	$\mathtt{C}^{\mathcal{I}} = \mathtt{D}^{\mathcal{I}}$
Définition de rôle	$\mathtt{R} \equiv \mathtt{S}$	$\mathtt{R}^{\mathcal{I}} = \mathtt{S}^{\mathcal{I}}$
Inclusion de concept	$\mathtt{C} \sqsubseteq \mathtt{D}$	$\mathtt{C}^{\mathcal{I}} \subseteq \mathtt{D}^{\mathcal{I}}$
Inclusion de rôle	$\mathtt{R} \sqsubseteq \mathtt{S}$	$\mathtt{R}^{\mathcal{I}} \subseteq \mathtt{S}^{\mathcal{I}}$
Assertion de concept	$\mathtt{C}(\mathtt{a})$	$\mathtt{a}^{\mathcal{I}} \in \mathtt{C}^{\mathcal{I}}$
Assertion de rôle	$\mathtt{R}(\mathtt{a},\mathtt{b})$	$(\mathtt{a}^{\mathcal{I}}, \mathtt{b}^{\mathcal{I}}) \in \mathtt{R}^{\mathcal{I}}$

TAB. 2.5 – Syntaxe et sémantique associées aux axiomes terminologiques et assertionels en LD

La *ABox* quant à elle représente un état particulier du domaine décrit par la *TBox* . Elle est constituée d'*axiomes assertionnels* qui adoptent la forme soit d'*assertions de concepts* à l'aide d'individus, soit d'*assertions de rôles* à l'aide de paires d'individus.

Définition, spécialisation, et assertion Les *axiomes terminologiques* (*i.e.* contenus dans la *TBox*) sont de deux formes.

- Les *égalités* de la forme générale $\mathtt{C} \equiv \mathtt{D}$ ($\mathtt{R} \equiv \mathtt{S}$) où C, D sont des concepts (et R, S des rôles). Les *définitions* de concepts (et de rôles) sont des égalités particulières de la forme $\mathtt{A} \equiv \mathtt{C}$ ($\mathtt{Q} \equiv \mathtt{R}$) où A est un concept atomique et C une description de concept (et Q un rôle atomique et R une description de rôle). L'axiome (Ax4) dans la BC du Tableau 2.4 est un exemple de définition de concept.
- Les *inclusions* ou *subsomption* de la forme générale $\mathtt{C} \sqsubseteq \mathtt{D}$ ($\mathtt{R} \sqsubseteq \mathtt{S}$) où C, D sont des concepts (et R, S des rôles). Les *spécialisations* de concepts (et de rôles) sont des inclusions particulières dont la partie gauche est un concept (un rôle) atomique de la même façon que pour les définitions. Cette spécialisation est quelque peu différente de la spécialisation des RCO puisque celle-ci signifie simplement que tout individu appartenant à l'interprétation de C appartient également à l'interprétation de D. Ainsi l'axiome (Ax3) dans la BC exemple est une spécialisation.

Les *axiomes assertionnels* (de la *ABox*) peuvent être de deux types différents selon qu'il s'agisse de l'assertion d'un concept ou d'un rôle :

- une *assertion de concept*, notée $\mathtt{C}(\mathtt{a})$, statue sur l'appartenance[27] d'un individu a au concept C[28], comme c'est par exemple le cas pour l'individu `cureDAntibiotique` qui instancie le concept `TraitementMédicamenteux` selon l'axiome (Ax6) de la BC Tableau 2.4 ;
- une *assertion de rôle*, notée $\mathtt{R}(\mathtt{a},\mathtt{b})$, statue sur le fait que b est relié à l'individu a par la relation R. De cette façon l'axiome (Ax7) Tableau 2.4 indique que l'individu `adrien` est traité par un individu appelé `cureDAntibiotique`.

Le Tableau 2.5 représente la sémantique associée aux différents axiomes d'une BC en LD. Les axiomes constituent en un sens l'élément de base de représentation d'une connaissance, à ce titre nous considérons dans le cadre des LD un axiome comme une **unité de connaissance**.

Mécanismes de raisonnement Si le rôle d'une BC en LD se limite au stockage des *TBox* et *ABox*, son principal avantage est de pouvoir être associée à des mécanismes de raisonnement. Ces mécanismes s'appuyent sur les deux premières opérations suivantes qui servent de briques de bases aux suivantes :

- le test de *subsomption* qui vérifie qu'un concept C subsume un concept D noté $\models \mathtt{D} \sqsubseteq \mathtt{C}$. Ainsi sur la BC prise en exemple la réponse au test de subsomption suivant $\models \mathtt{PatientSousTraitement} \sqsubseteq$

[27] Par analogie avec les langages de RCO on parle également d'instanciation.

[28] Pour être tout à fait exact il faudrait dire "l'interprétation de a qui appartient à l'interprétation de C".

`Patient` est vrai. Cette subsomption n'est pas explicitement écrite dans la BC . Cependant la définition de l'axiome (Ax4) signifie que toute instance du concept `PatientSousTraitement` est également instance du concept `Patient` (ainsi que du concept $\exists$ `estTraité`.`TraitementMédicamenteux`), ce qui permet aux mécanismes de raisonnement de déduire la réponse.

- Le test de *satisfiabilité* qui vérifie qu'un concept peut admettre des instances.
- La *classification des concepts* qui permet de déterminer la position relative de chaque concept dans la hiérarchie de concepts.
- La *classification d'instances* qui permet de déterminer pour un individu, les concepts dont il est instance. Suivant ce mécanisme il est possible de déterminer sur la base de la BC exemple que l'individu `adrien` est également instance du concept `PatientSousTraitement` ce qui n'est pas explicitement décrit. En effet l'instance `adrien` remplit l'ensemble des *conditions nécessaires et suffisantes* à l'appartenance à ce concept, *i.e.* en termes informels être un patient et être traité par quelque chose qui est un traitement médicamenteux.
- La *recherche d'instances* (ou *instance retrieval* en anglais) qui permet de déterminer pour un concept l'ensemble des individus qui en sont instances.

L'efficacité de certains mécanismes de raisonnement, plus complexes, est conditionnée par la LD choisie. Parmi ceux là nous citerons

- la *recherche du concept le plus spécifique* (ou *most specific concept*) qui consiste à déterminer pour un concept (ou un individu) quel est le concept le plus spécifique qui le subsume (ou quel est le concept le plus spécifique dont il est instance).
- la *recherche du subsumant commun le plus spécifique* (ou *least common subsumer*) qui recherche le concept le plus spécifique qui subsume en même temps deux concepts donnés (ou dont deux individus donnés sont instances).

L'utilisation de ces derniers mécanismes de raisonnement plus complexes est discutée dans [BCM+03].

L'effervescence autour du Web Sémantique et l'adoption pour ce dernier d'un langage standard (le OWL présenté dans la section 2.3.1) contenant une LD ont favorisé les travaux de recherche et les avancées en LD. Malgré leur manque de convivialité, les logiques de descriptions constituent un moyen de représenter les connaissances actuellement préféré aux langages de RCO. Cependant les RCO présentent des avantages qui pourraient inspirer des évolutions des LD par exemple en ce qui concerne les méthodes de raisonnement telles que l'inférence de valeur.

Pour plus de détails sur la comparaison entre RCO et LD nous conseillons la référence [Duc00] de Ducourneau *et al.* .

2.3 Ontologies et Bases de Connaissances

Le terme *ontologie* est un emprunt à la philosophie au sein de laquelle l'ontologie est une branche de la métaphysique dédiée à l'étude des propriétés de ce qui *est*, de ce qui *existe*.

En informatique une *ontologie* est une *représentation de connaissances*. Cependant, la notion d'ontologie est utilisée pour désigner différentes formes de représentation de connaissances. Ceci est particulièrement vrai en bioinformatique où le terme d'ontologie est utilisé selon différentes considérations [GW04]. Ainsi pour certains une ontologie peut se limiter à un *vocabulaire contrôlé*, *i.e.* une liste de termes consensus en rapport avec un domaine. Ce peut être un vocabulaire contrôlé associé à une *hiérarchie* comme c'est le cas pour la GENE ONTOLOGY [ABB+00]. Il peut également être associé à ces vocabulaires des listes de synonymes qui permettent de mettre en correspondance un terme arbitraire avec le terme choisi comme référence. De façon plus complexe et aussi plus complète, une ontologie peut être une représentation des *concepts* d'un domaine ainsi que des relations qui existent entre ces concepts. Alors la notion de *concept* représente un ensemble fini ou infini, ainsi par exemple le concept de protéine

représente (intuitivement) l'ensemble des protéines.

C'est à cette dernière forme d'ontologie que nous nous référons dans cette thèse, en accord avec la définition de Gruber pour qui une ontologie est

"*une spécification formelle et explicite d'une conceptualisation partagée*" [Gru93].

Les concepts et leurs relations représentés dans une ontologie peuvent être définis de façon plus ou moins précise selon le formalisme (l'ensemble de symboles et de règles de syntaxe) utilisé pour les décrire. L'utilisation de langages de représentation des connaissances permet d'associer aux concepts et aux relations une description formelle qui fait référence à une sémantique clairement définie dans le cas des LD (voir la section 2.2). L'avantage de l'utilisation d'une telle sémantique est de pouvoir associer aux concepts et relations de l'ontologie une interprétation unique qui puisse ainsi être comprise de la même façon par deux humains, ou par un humain et une machine. Ce point est important dans la représentation des connaissances biologiques, car d'une part, il est nécessaire pour un utilisateur de comprendre le modèle biologique exprimé, et d'autre part, il est important que les entités biologiques représentées puissent être exploitées par des programmes bioinformatiques.

REMARQUE : Le fait qu'une ontologie soit associée à une seule interprétation ne veut pas dire que pour un domaine il n'existe qu'une seule conceptualisation et qu'une seule interprétation admissibles. Au contraire, un domaine peut donner lieu à plusieurs interprétations qui peuvent alors mener à la création d'ontologies différentes. Les ontologies alors coexistantes reflètent les différentes perspectives qui existent sur le domaine en question. Par exemple, le domaine de la pharmacogénomique peut être conceptualisé selon la perspective des cliniciens ou celle des biologistes moléculaires. Pour les premiers la pharmacogénomique est considérée du côté de la médecine personnalisée et des relations entre un diagnostic génétique, un traitement médicamenteux, et un phénotype macroscopique (une pression artérielle élevée par exemple). Pour les seconds, la pharmacogénomique est considérée à un niveau moléculaire impliquant notamment les relations entre un groupe de SNP, une molécule (le principe actif du médicament), et un phénotype moléculaire (la modulation du taux d'expression d'un gène par exemple).

De façon formelle nous définissons une ontologie, d'une façon similaire à [ES07], comme suit :

Définition 2.19 (Ontologie) *Une ontologie O est un système de symboles (S_c, S_r, H, A) consistant en :*
- *un ensemble S_c de concepts, et un ensemble S_r de relations binaires (D,R), entre deux concepts D,$R \subset S_c$ appelés le domaine et le co-domaine (*domain *et* range *en anglais) ;*
- *une hiérarchie H, où les concepts et relations sont hiérarchiquement reliés par la relation de **subsomption**, i.e. une relation d'ordre partiel noté $\sqsubseteq$, où $C_1 \sqsubseteq C_2$ signifie que C_1 est un sous-concept de C_2, et $r_1 \sqsubseteq r_2$ signifie que r_1 est une sous-relation de r_2 ;*
- *un ensemble d'axiomes A qui décrivent des contraintes sur les concepts et les relations.*

Les ontologies auxquelles nous ferons allusion dans la suite de cette thèse sont des ontologies représentées en LD. Or en LD le terme ontologie est traditionnellement peu employé. Les notions de *TBox* et *ABox* clairement définies lui sont préférées. Pour cela il est important de préciser que dans cette thèse, une *ontologie* en LD correspond à une *TBox* , alors qu'une *Base de Connaissance* (BC), pour sa part, fait référence à l'ensemble *TBox – ABox*.

2.3.1 OWL et le Web sémantique

Le Web sémantique est d'abord une idée ou une vision du Web selon laquelle le contenu des ressources diffusées sur le Web est rendu accessible aux programmes informatiques de façon à ce que ceux-ci soient mieux à même de répondre aux besoins des utilisateurs humains [BLHL01]. Il s'agit de décrire ces

ressources, ou plutôt les données qu'elles contiennent, selon une représentation formelle, c'est à dire en lien avec une sémantique clairement définie et conçue pour être interprétée par des programmes. Ceux-ci pourraient alors manipuler sous forme de connaissances les données disponibles sur le Web pour découvrir des connaissances implicites ou nouvelles via des mécanismes de raisonnement. A la base de l'infrastructure du Web sémantique se trouvent les *ontologies*. Celles-ci apportent les éléments essentiels qui permettent l'introduction des données du Web dans un contexte à base de connaissances.

OWL (*Web Ontology Language*) est le langage choisi comme standard par le W3C[29] pour la diffusion des ontologies sur le Web et constitue en ce sens la principale technologie sur lequel repose le Web sémantique. OWL s'appuie à la fois sur les technologies du Web (comme HTML, XML et RDF) et sur des langages de représentation des connaissances tels que les systèmes de RCO et les LD.

La spécification initiale de OWL reposait sur les exigences suivantes :

– le langage doit être associé à une sémantique standard et formellement définie permettant la mise en œuvre de mécanismes de raisonnement maîtrisés,
– le langage doit être très expressif pour prendre en compte la variété des domaines et des applications envisagés dans le cadre du Web sémantique.

Ces deux éléments expliquent en partie le choix des LD pour représenter les connaissances en OWL. L'Annexe B propose une correspondance entre les constructeurs de LD et les constructeurs OWL. De la même façon qu'il existe plusieurs sous-familles de LD, il existe différents *profils* OWL (OWL-Lite, OWL-DL, et OWL Full en sont les trois principaux) dont les différences résident dans les constructeurs qu'il proposent. Par exemple, le profil OWL-DL propose un ensemble de constructeurs qui correspond à la logique $\mathcal{SHOIN}(\mathcal{D})$.

Le langage OWL est difficile à écrire et lire directement, il est donc plus généralement développé et édité à travers des éditeurs d'ontologie ou de BC comme Protégé [KFNM04] ou Swoop [KPS+06]. Nous proposons en Annexe C le code OWL qui correspond à la BC représentée dans le Tableau 2.4.

FaCT++ [TH06], Pellet [SP04], et RacerPro [HM03] sont des logiciels qui permettent de mettre en œuvre les mécanismes de raisonnement standards en LD sur une ontologie (ou une BC) implantée en OWL.

2.3.2 Construction d'ontologies

De nombreuses méthodes pour le développement d'ontologies ont été proposées [UK95, FGPJ97, NM01]. Nous ne cherchons ici ni à les passer en revue, ni à les comparer, mais plutôt à faire ressortir les opérations importantes à mettre en œuvre lors de la construction, manuelle ou semi-automatique, d'ontologies. Pour une vue d'ensemble des méthodes de construction d'une ontologie, nous orientons le lecteur vers le chapitre 3 du livre de Gómez-Pérez [GPCGFL03].

L'ensemble de ces méthodes s'inspire du génie logiciel comme l'illustre le cycle de vie d'une ontologie proposé par Dieng *et al.* [DCGR98] et représenté Figure 2.6 qui met en avant le côté *itératif* de la construction ainsi que ses principales étapes.

La construction d'une ontologie est un processus *collaboratif* où les experts du domaine (et éventuellement des systèmes d'apprentissage) doivent être fortement impliqués. Nous nous intéressons plus particulièrement aux étapes de *spécification* des besoins, de *conception*, et d'*évaluation* de l'ontologie.

Spécification Cette étape consiste à définir, en étroite collaboration avec les experts du domaine, le *domaine* et l'*objectif* de l'ontologie.

Concernant le *domaine*, il s'agit de préciser d'abord le domaine de connaissances que l'ontologie doit représenter, mais aussi avec quel niveau de *granularité* celui-ci doit être représenté. Ainsi, pour

[29] *World Wide Web Consortium* : consortium international pour la standardisation et la promotion des technologies du Web, `http://www.w3.org/`

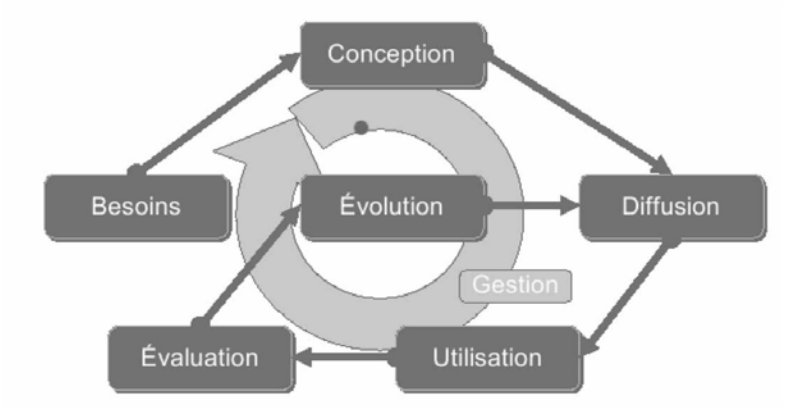

FIG. 2.6 – Cycle de vie d'une ontologie. *Source* : [DCGR98].

la création d'une ontologie des protéines, il est important de préciser clairement les limites du domaine à représenter : par exemple les protéines humaines ou les protéines phosphatases (humaine et non-humaine). Le niveau de granularité souhaité doit alors préciser le niveau de détail attendu dans la représentation du domaine. Pour une ontologie des protéines il faut spécifier par exemple que seules leurs annotations fonctionnelles et leur masse est importante ou alternativement qu'une description plus fine est nécessaire impliquant la représentation de la composition en acides aminés, des groupements fonctionnels de ces acides aminés et de leurs propriétés physico-chimiques.

La définition de l'*objectif* de l'ontologie doit déterminer les applications pour lesquelles l'ontologie est construite. Savoir à quoi va servir l'ontologie est déterminant pour déterminer les concepts à représenter et le point de vue selon lequel les représenter. Ainsi cette opération doit permettre de déterminer si notre ontologie des protéines doit servir à l'annotation de bases de données, à la classification de protéines, ou encore à l'extraction de connaissances relatives aux conséquences fonctionnelles des variations génomiques. Pour formaliser les objectifs et pour permettre l'évaluation de l'ontologie, il est possible de définir des *questions de compétence* qui sont des exemples concrets de questions auxquelles l'ontologie doit permettre de répondre [Gan05]. Vérifier que l'ontologie, une fois construite, y répond correctement est un critère d'évaluation de cette dernière.

Une opération importante de la phase de spécification est le recensement systématique des ressources de données ou de connaissances déjà existantes. Ces ressources peuvent être des sources de données, des schémas de sources, des vocabulaires contrôlés, des ontologies de domaines incluses, recouvrant, ou chevauchant le domaine considéré. Ce recensement a deux buts. Le premier est d'avoir une vue globale des données, et connaissances manipulées dans le domaine de sorte à considérer leur représentation dans l'ontologie. Le second est de réutiliser au maximum l'existant. Pour reprendre notre exemple d'une ontologie des protéines, l'analyse des données et des schémas des bases de données UniProt[30], PDB[31] donne une vue sur un certain nombre de propriétés des protéines. Aussi la considération de l'ontologie appelée *PRotein Ontology*[32] (PRO) et de l'ontologie *Phosphatase Ontology*[33] permet de s'inspirer ou de réutiliser les représentations existantes des connaissances.

Cette étape est également le moment approprié pour l'adoption de règles de nomenclature strictes pour nommer les concepts et rôles de l'ontologie.

[30]Universal Protein Resource : `http://www.uniprot.org/`
[31]Protein Data Bank : `http://www.rcsb.org/`
[32]`http://pir.georgetown.edu/pro/`
[33]`http://www.bioinf.manchester.ac.uk/phosphabase/`

Conception La conception de l'ontologie comprend trois opérations principales :

- La *conceptualisation* : elle consiste en l'identification des concepts du domaine et des relations entre ces concepts. Elle peut commencer par la définition de listes de termes propres au domaine, termes qui serviront à l'identification et à la définition des concepts, de leurs relations et de leur articulation avec des ontologies existantes.
- la *formalisation* : c'est la traduction de la conceptualisation dans un formalisme de représentation des connaissances, par exemple une LD.
- l'*implémentation* : il s'agit de coder l'ontologie formalisée en un langage de représentation des connaissances, par exemple OWL.

En pratique les étapes de formalisation et d'implémentation sont souvent menées de front.

Évaluation Classiquement l'étape d'évaluation fournit des mesures sur l'ontologie selon des critères structurels et fonctionnels [GCCL06]. L'idéal est que ces critères d'évaluation soient définis durant l'étape de spécification de l'ontologie. C'est typiquement le cas des questions de compétence auxquelles l'ontologie doit permettre de répondre.

Selon Dellschaft et Staab [DS08], les critères structurels et fonctionnels sont utilisés dans trois types d'évaluations :

- l'*évaluation rapportée à la tâche*, où l'on mesure essentiellement comment l'ontologie améliore la réalisation d'une tâche ;
- l'*évaluation rapportée au corpus*, où l'on mesure la capacité de l'ontologie à représenter les connaissances d'un domaine en se référant au contenu d'un corpus de documents représentatif du domaine ;
- l'*évaluation rapportée aux critères*, où ce sont des critères objectifs, le plus souvent d'ordre structural, qui sont mesurés.

Évidement l'évaluation rapportée à la tâche peut être considérée comme la plus importante, puisque c'est celle-ci qui donne la mesure de l'efficacité de l'ontologie dans la tâche à laquelle on l'a assignée. Cependant les deux autres types d'évaluations sont des moyens ponctuels d'évaluer certains aspects de l'ontologie dont dépendra forcement son efficacité à résoudre une tâche particulière.

Idéalement, l'évolution d'une ontologie est un processus continu qui suit de près l'évolution de l'état des connaissances qu'elle représente.

Le développement complet d'une ontologie, même semi-automatique, est un tâche longue qui mérite une grande attention. La contrepartie des efforts demandés pour son développement réside notamment dans la sémantique qu'elle fournit qui peut être utilisée par des mécanismes de raisonnement mais pas seulement. Les sections suivantes de ce chapitre présentent l'utilisation d'ontologies dans le cadre d'intégration de données (section 3), puis, plus généralement dans le cadre de l'ECBD (section 4).

3 Utilisation des ontologies pour l'intégration de données hétérogènes

L'objectif de cette section est d'introduire la notion d'*intégration sémantique*, *i.e.* d'intégration de données fondées sur l'utilisation de connaissances du domaine et de mécanismes de raisonnement. Au vu de cet objectif, nous n'entreprendrons pas un état de l'art exhaustif sur l'intégration de données mais nous nous focaliserons seulement sur deux systèmes concurrents (l'*approche entrepôt* et l'*approche médiateur*) parce qu'ils nécessitent la définition de *mapping* c'est à dire de mise en correspondance entre les données et qu'ils ont donné lieu à quelques systèmes opérationnels. Ces deux approches nous paraissent aujourd'hui les plus propices à supporter des approches à base de connaissance telles que celle qui fait l'objet du chapitre 3.

3.1 Les systèmes d'intégration de données

Les deux approches principales pour l'intégration de données se distinguent essentiellement par la localisation des données manipulées par le système [Hal01] :

- l'*intégration matérialisée* pour laquelle les données sont dans un *entrepôt de données* où elles sont rapatriées depuis leur source d'origine ;
- l'*intégration virtuelle* pour laquelle les données restent dans les sources d'origine où elles sont manipulées par le biais d'un *médiateur*.

3.1.1 L'intégration matérialisée ou entrepôt

L'approche matérialisée ou *entrepôt de données* consiste en la construction d'une base de données réelle appelée entrepôt pour stocker les données provenant de différentes sources. Les entrepôts de données sont souvent choisis dans l'industrie pour le support d'aide à la décision qu'ils constituent, notamment grâce à leur association aux techniques OLAP [AAD+96]. Un système d'intégration suivant une telle approche est constitué de trois parties représentées Figure 2.7 : l'*entrepôt* de données proprement dit, les *sources* de données, et les *magasins* de données.

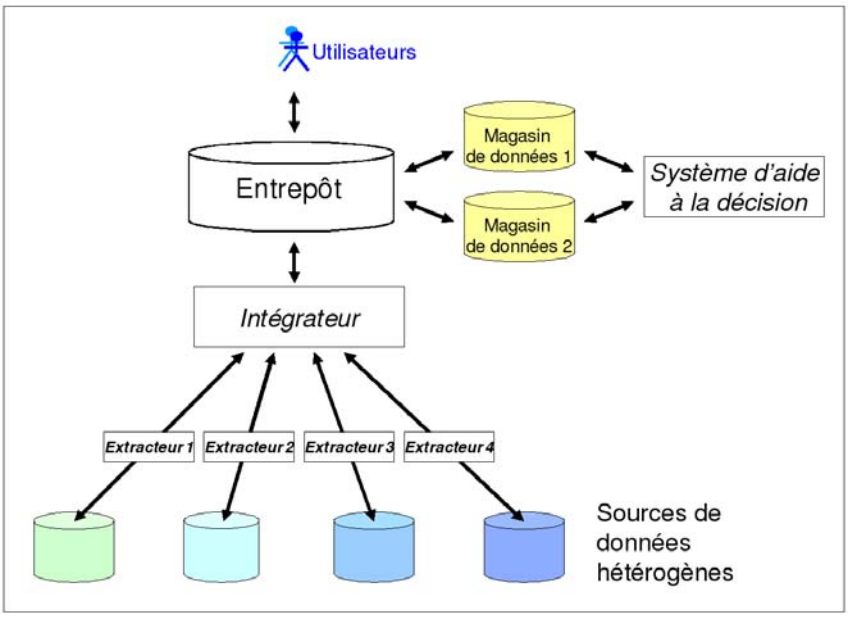

FIG. 2.7 – Architecture d'un système d'intégration de données suivant l'approche entrepôt

Dans une approche entrepôt de données, l'intégration s'appuie sur un schéma global défini pour l'entrepôt. Les données sont extraites des sources, transformées en un format de représentation compatible avec le schéma de l'entrepôt, éventuellement filtrées, et insérées dans l'entrepôt par des *extracteurs*. L'utilisateur peut interroger directement l'entrepôt en utilisant les techniques classiques d'interrogation de base de données. L'utilisateur peut également interagir avec l'entrepôt par l'intermédiaire des magasins de données dont l'objectif est de proposer des *vues* particulières sur les données qui pourront faciliter leur analyse dans un processus d'aide à la décision. L'entrepôt GEDAW est un exemple de mise en œuvre de l'approche entrepôt pour l'intégration et l'analyse de données relatives au transcriptome humain [GMB+05]. Le système BIOMART et BIOWAREHOUSE sont des systèmes plus génériques dédiés à l'intégration de données biologiques suivant une approche entrepôt [KKS+04, KLW08].

Une limite principale à cette approche réside dans le développement nécessaire de méthodes d'extraction et d'intégration des données capables de rafraîchir périodiquement le contenu de l'entrepôt, tout en tenant compte de la mise à jour des sources.

3.1.2 L'intégration virtuelle ou médiateur

Dans l'approche *médiateur*, l'intégration de données est fondée sur la définition d'un schéma global unifiant les schémas hétérogènes des sources à intégrer. La description d'un tel schéma implique la mise au point de *mappings* (parfois présentés sous la forme de *vues*). Un mapping est la description du contenu d'une source dans le vocabulaire unique imposé par le schéma global. L'architecture médiateur classique s'articule suivant trois niveaux représentés Figure 2.8 : le *médiateur*, les *wrappers*, et les *sources*. Au niveau du médiateur, le schéma global fournit un vocabulaire unique qui sert à *(1)* l'expression des requêtes de l'utilisateur, et *(2)* à la définition de mappings, *i.e.* la description du contenu de chaque source. Les wrappers (également appelés *adaptateurs* pour éviter l'anglicisme) s'appuyent sur la définition des mappings pour *(a)* traduire les requêtes exprimées dans les termes du vocabulaire du schéma global en des requêtes exprimées selon le vocabulaire des sources ; *(b)* traduire les réponses aux requêtes locales (*i.e.* sur les sources) en des réponses compatibles avec le schéma global du médiateur.

La constitution d'un mapping se fait par la définition de multiples mises en correspondances entre les relations (au sens des bases de données relationnelles) du schéma global et les relations du schéma local. Ces mises en correspondance peuvent être décrites suivant deux approches différentes [Len02]. La première approche est appelée *Global As View* (ou GAV) selon laquelle les relations du schéma global sont exprimées en fonction des relations du schéma local. La seconde est l'approche *Local As View* (ou LAV) où inversement, dans un premier temps un schéma global est défini de façon indépendante, puis au niveau local les relations des schémas locaux sont reformulés dans les termes du schéma global. Le contenu des sources est décrit par un ensemble de mappings sur les relations du schéma global.

Selon Lenzerini [Len02], la description d'un système d'intégration de données peut être formalisée selon un triplet $(\mathcal{G}, \mathcal{S}, \mathcal{M})$ regroupant ses trois composants principaux :

- le schéma global $\mathcal{G}$,
- les schémas des sources $\mathcal{S}$, et
- le mapping $\mathcal{M}$ entre $\mathcal{G}$ et $\mathcal{S}$ décrit par un ensemble de correspondances de la forme suivante

$$q_{\mathcal{G}} \rightsquigarrow q_{\mathcal{S}} \text{ ou}$$
$$q_{\mathcal{S}} \rightsquigarrow q_{\mathcal{G}}$$

où $q_{\mathcal{G}}$ et $q_{\mathcal{S}}$ sont deux requêtes respectivement sur le schéma global et sur le schéma des sources.

La tâche du médiateur consiste à reformuler, à l'aide des mappings, les requêtes qui lui sont posées dans les termes du schéma global en des requêtes exprimées dans les termes des schémas des sources

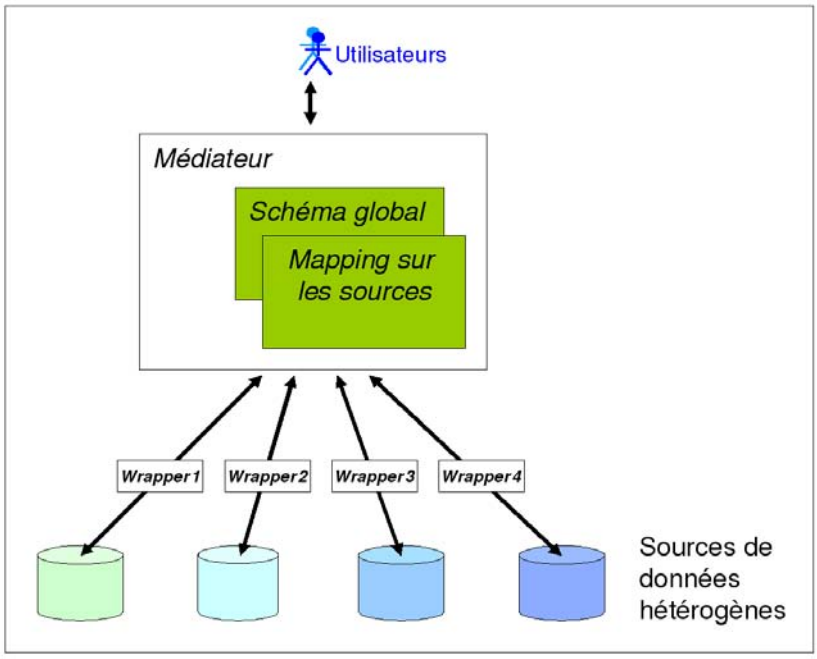

FIG. 2.8 – Architecture d'un système d'intégration de données suivant l'approche médiateur

de données et inversement. Cette tâche de reformulation est appelée la *réécriture de requêtes*. Le projet PICSEL constitue un exemple d'architecture médiateur originale notamment parce qu'elle propose une approche *hybride* GAV-LAV, ainsi qu'un schéma global exprimé suivant un formalisme de représentation des connaissances appelé CARIN [RFG+02]. Le travail de Mougin *et al.* [MBB+08] est une illustration de l'approche médiateur appliquée à des bases de données biologiques et dont la particularité est de proposer des méthodes semi-automatiques pour faciliter la définition des mappings nécessaires.

Les limites de l'approche médiateur résident, suivant une approche GAV, principalement dans la mise à jour du schéma global lors de l'intégration d'une nouvelle source ou, suivant une approche LAV, dans la réécriture des requêtes. Que l'approche adoptée soit GAV ou LAV, le travail le plus important demeure premièrement la définition des mappings qui peut demander l'intervention d'un expert du domaine d'application, et deuxièmement la conception des wrappers toujours dépendante du type de source considéré.

L'approche entrepôt présente deux avantages principaux. Le premier est lié au fait que les données intégrées sont à disposition au sein de l'entrepôt spécifiquement créée, ce qui permet de les transformer et les adapter directement et facilement à l'utilisation que l'on souhaite en faire. Le second avantage est lié au fait que les données soient regroupées dans une seule source, ce qui facilite l'exploitation du système.

Cependant, comparée à l'approche médiateur, l'approche entrepôt présente une contrainte majeure : les réponses aux requêtes ne proviennent pas directement des sources, mais des données matérialisées au sein de l'entrepôt. Ceci implique la mise à jour de l'entrepôt à chaque modification des sources, or dans certains domaines comme les sciences du vivant, les sources sont parfois soumises à une évolution hebdomadaire voir quotidienne.

REMARQUE : Les architectures orientés services (ou *SOA* pour *Service Oriented Architecture* en anglais) sont des formes particulières d'architecture médiateur dont les différents composants sont associés à des composants logiciels (*i.e.* les services) qui interagissent entre eux. La communication entre les différents services s'appuie sur un vocabulaire partagé qui peut être comparé à un schéma global. L'utilisation

d'ontologie pour la description du schéma global peut faciliter la découverte de services ainsi que leur utilisation dans la définition de workflows. Les articles [MD07a, MD07b, DPW08] peuvent servir d'introduction à cette problématique.

L'importance prise par le Web a conduit notamment au développement de nombreuses déclinaisons et variantes des approches d'intégration médiateur et entrepôt [Hac04]. Un exemple notable est le développpement d'approches adaptées à la structure particulière du Web qui pour cela utilise une architecture *paire à paire* [CGLR04, RAC+06].

3.2 Problème d'hétérogénéité et intégration sémantique

3.2.1 Hétérogénéité de données et de schémas

La problématique première de l'intégration de données est l'*hétérogénéité des données* et l'*hétérogénéité des schémas* qu'il est nécessaire de résoudre pour mettre en correspondance les sources et autoriser l'interrogation et la réponse aux requêtes de façon transparente [Hal05, Saï07].

L'*hétérogénéité des données* est due au fait que deux bases de données n'utilisent pas le même vocabulaire ou référentiel pour représenter une même donnée. Par exemple, le fait qu'un nucléotide, clairement positionné sur le génome humain, puisse être soit une Adénine (A) soit une Guanine (G) selon les individus est noté "A/G" dans la base de données dbSNP. Cependant il existe un référentiel différent, le code IUPAC[34], utilisé dans d'autres bases de données biologiques selon lequel le fait qu'un nucléotide puisse être soit une Adénine soit une Guanine est simplement noté par la lettre R (pour faire référence aux Purines).

L'*hétérogénéité des schémas* provient quant à elle du fait que deux bases de données peuvent proposer deux conceptualisations différentes d'une même entité. Cela peut correspondre à l'utilisation de noms d'attributs différents. C'est par exemple le cas pour l'attribut faisant référence à l'alternative possible entre deux nucléotides pour un variant génomique qui est nommé "Allele" dans la dbSNP et "Variation" dans la base PharmGKB. De façon plus complexe, l'hétérogénéité peut résider dans la notion même de variant génomique qui peut diverger entre deux sources. C'est justement le cas pour dbSNP et Uniprot qui considèrent, respectivement, un variant soit comme une alternative entre deux nucléotides pour une même position sur une séquence d'ADN, soit comme une alternative entre deux acides aminés pour une position sur une séquence protéique.

Les ontologies peuvent contribuer à la résolution du problème d'hétérogénéité des données et des schémas. En effet, elles permettent la description formelle des concepts d'un certain domaine ainsi que des relations existant entre ces concepts. Un utilisateur ou un concepteur peut décrire une donnée, une relation présentes dans une source grâce à une définition formelle à laquelle est associée une sémantique clairement établie. Ensuite, il peut exploiter cette définition pour intégrer (ou seulement partager) de façon non ambiguë le contenu de la source en question. Définitions formelles et sémantique peuvent en pratique être représentées sous la forme d'axiomes logiques composant une ontologie, c'est pourquoi on parle d'approche d'*intégration fondée sur une ontologie* ou d'*intégration sémantique*.

3.2.2 Enjeux de l'intégration sémantique

Nous discernons cinq problèmes dont les résolutions constituent les principaux enjeux pour la mise au point d'un système d'intégration de données sémantique [PLC+08] :

1. Permettre la gestion de *grands volumes* de données en utilisant la représentation formelle d'une ontologie. En effet, il existe un fossé entre l'échelle des systèmes de gestion de bases de données

[34] `http://www.dna.affrc.go.jp/misc/MPsrch/InfoIUPAC.html`

capables de gérer efficacement des quantités de données importantes et l'échelle des systèmes à base de connaissance pour qui gérer une ontologie ou une BC trop volumineuse reste problématique.

2. Permettre des *requêtes sur les données* selon le langage de représentation et les termes de l'ontologie. Ceci implique la mise en correspondance entre le langage de représentation des connaissances et un langage de requête adapté aux sources.
3. *Choisir le langage* de représentation des connaissances. Plus un langage est expressif, plus il sera complexe de raisonner sur ce langage. Il faut donc trouver un compromis entre une expressivité suffisante pour représenter correctement le domaine et une complexité compatible avec les besoins en raisonnement liés à la réponse aux requêtes.
4. Décrire le *mapping données-ontologie*. Cela revient à mettre en correspondance les données des sources avec les instances des concepts et rôles de l'ontologie, ou en d'autres termes à relier la sémantique implicite des données à la sémantique explicite de l'ontologie.
5. Résoudre le *problème d'impédance*. Le problème d'impédance (*impedance mismatch* en anglais) réside dans le fait qu'il n'existe pas une correspondance exacte entre la façon dont sont représentées les informations dans une base de données relationnelle (par la paire attribut-valeur) et la façon dont elle peuvent l'être en terme d'objets (par la paire classe-objet) et donc d'instances de concepts dans une ontologie. Il en résulte le besoin de mécanismes capables de faire correspondre valeurs et objets.

Les réponses aux problèmes 1 et 2 sont en partie apportées par des solutions technologiques issues de travaux sur le Web sémantique. Ainsi différents outils et méthodes permettent de gérer des BC volumineuses notamment en permettant leur enregistrement dans des systèmes de gestion de bases de données relationnelles. SESAME [BKvH02], INSTANCE STORE [HLTB04] en sont des exemples, et les résultats du challenge intitulé "Billion triplet challenge"[35] laisse présager des solutions techniques prometteuses. Concernant les requêtes sur les BC, notons que le langage SPARQL commence à émerger parmi les diverses propositions antérieures (OWL Abstract Syntax[36] par exemple) puisqu'il fait l'objet d'une recommandation du W3C depuis janvier 2008[37].

Concernant le 3$^{\text{ième}}$ problème, la communauté des LD a cherché à comparer les langages de représentation d'une BC pour montrer lequel pouvait être le plus adapté à un accès à de grands volumes de données. Les travaux de Hustadt *et al.* et Calvanese *et al.* montrent notamment qu'une requête[38] même sur une BC exprimée en un profil simple du OWL (OWL-Lite et OWL-DL) est trop complexe (co-NP complexe) pour être envisageable sur de larges volumes de données [HMS05, CGL$^+$06]. Une solution proposée pour garantir la réponse aux requêtes dans un temps fini (polynômial) est l'utilisation de logiques de descriptions restreintes (*i.e.* à l'expressivité restreinte) comme par exemple $\mathcal{EL}^{++}$ [BBL05]. Dans cette même idée, le W3C travaille notamment au développement d'un profil de OWL appelé OWL-R[39], moins expressif que OWL-Lite, à partir duquel la recherche d'instance pourrait être facilitée.

Les travaux réalisés dans l'optique de résoudre les problèmes 4 et 5 relatifs à la définition d'un mapping données–connaissances sont présentés dans la section suivante.

3.3 Le mapping données–connaissances

L'utilisation d'une ontologie pour l'intégration de données est possible uniquement à partir du moment où il existe un *mapping entre données et connaissances*, *i.e.* un ensemble de mises en correspon-

[35] http://www.mindswap.org/blog/2007/12/05/announcing-the-open-web-billion-triple-challenge-iswc-08/
[36] http://www.w3.org/TR/owl-semantics/
[37] http://www.w3.org/TR/rdf-sparql-query/
[38] Une requête en LD est le plus souvent considérée comme un mécanisme de raisonnement de *recherche d'instance*.
[39] http://www.w3.org/TR/2008/WD-owl2-profiles-20080411/#OWL-R_Full

dance entre des données d'une source d'un côté, et les instances des concepts et rôles d'une ontologie de l'autre.

CARIN introduit par Rousset *et al.* [RFG+02], ou R_2O introduit par Barrasa *et al.* [BCGP04], sont des propositions de langages spécifiques pour la description de mappings données–ontologie.

Cependant ces approches ne prennent pas en considération le problème d'impédance entre valeurs et objets. Ce problème requiert la définition de mécanismes capables de faire correspondre les valeurs des données aux objets de l'ontologie et notamment de préciser comment les identifiants des objets peuvent être construits à partir des valeurs de données. Ce genre de mécanismes a par contre été décrit dans le cadre d'approches entrepôt faisant intervenir des bases de données objets [HY90, CGL+01]. Il s'agit alors de définir des fonctions symboliques (par exemple de conversion) et de les associer à une liste d'attributs à considérer pour construire l'identifiant de l'objet correspondant. Le même genre de fonction est défini pour réconcilier des données hétérogènes issues de diverses sources et permettre une intégration dans une représentation homogène.

De plus, des formalismes comme $\mathcal{SHOIN}(\mathcal{D})$ ou *DL-Lite*$_A$ permettent d'associer aux instances de concepts des valeurs [HPSvH03, CGL+07]. Par exemple la logique $\mathcal{SHOIN}(\mathcal{D})$ qui est la logique sur laquelle s'appuie le profil OWL-DL de OWL permet la manipulation des concepts particuliers qui correspondent aux types de données (le $\mathcal{D}$ signifie *datatype* en anglais). Ainsi associer une instance à une valeur revient en $\mathcal{SHOIN}(\mathcal{D})$ à instancier un rôle associant cette instance et une instance du type de données (entier, chaîne de caractère, etc.) auquel correspond la valeur en question.

Le travail récent de Poggi *et al.* utilise les outils présentés dans cette section (langage formel pour la description de mapping, fonction de conciliation valeur-objet, LD manipulant des valeurs) pour décrire de façon théorique un système d'interrogation de données fondé sur une ontologie [PLC+08]. Nous nous sommes basés sur cette approche théorique et l'avons adaptée de sorte à la rendre opérationnelle et à l'accorder à nos objectifs d'intégration de données dans le contexte d'une BC. Ainsi, nous proposons dans le chapitre 3 une approche originale d'intégration de données qui s'inspire de l'approche médiateur dont l'objectif principal n'est pas la réponse à une requête mais le peuplement d'une BC.

3.4 Utilisation des ontologies en bioinformatique : intégration de données et plus si affinités

L'utilisation principale des ontologies en bioinformatique est l'intégration de données, mais ce n'est pas la seule. Ainsi cette section présente non seulement l'utilisation des ontologies pour l'intégration de données en bioinformatique mais aborde également leurs autres applications, toutes relativement connexes à l'intégration.

Dans une revue récente, Daniel Rubin *et al.* recensent les utilisations des ontologies en bioinformatique selon six catégories [RSN07] :

- la représentation de connaissances encyclopédiques,
- le Traitement Automatique des Langues (TAL),
- la recherche et l'interrogation de données biomédicales hétérogènes,
- l'échange de données entre applications,
- l'intégration de données, et
- l'utilisation de mécanismes de raisonnement.

Les sections suivantes illustrent ces différentes applications.

3.4.1 La représentation de connaissances encyclopédiques

De nombreuses ontologies en biologie sont partagées sur le Web via des portails dédiés comme le site de l'OBO Foundry[40] ou le BioPortal[41] [SAR+07, RMKM08]. Ceci permet à des personnes de réutiliser des ontologies sans avoir à construire celles-ci au préalable. Cependant la construction reste la phase préliminaire indispensable à toute utilisation ou réutilisation d'une ontologie. La richesse des connaissances disponibles dans certains domaines, comme l'anatomie humaine par exemple, la complexité d'autres, comme l'épigénomique, ou encore la co-existence de plusieurs théories pour un même domaine, comme la psychiatrie, imposent l'utilisation de méthodologies rigoureuses et parfois le développement d'outils particuliers (*e.g.* des outils collaboratifs) pour la représentation des connaissances en biologie de façon encyclopédique ce qui en fait une discipline à part entière. Les efforts de développement mis en œuvre notamment pour l'ontologie FMA (*Foundational Model of Anatomy*) disponible sur les portails cités précédemment ou ceux mis en œuvre pour le développement de l'ontologie NeuroWeb illustrent des méthodes et outils spécialement développés [RMM+98, CMF+07].

En plus d'héberger et de partager des bio-ontologies les initiatives OBO Foundry et du BioPortal participent à leur développement. L'OBO Foundry milite pour favoriser le suivi de standards de qualités dans le développement des bio-ontologies [Fou08]. Le BioPortal, et plus particulièrement sa version 2.0[42], propose un ensemble d'outils pour faciliter la navigation dans les ontologies, le développement collaboratif, la définition et le partage de mappings. Notons que si de nombreuses bio-ontologies sont de simples taxonomies ou des vocabulaires contrôlés, les résultats de travaux récents permettent de les transformer en OWL [Hor07, AEB+08].

3.4.2 Le Traitement Automatique des Langues

Les ontologies sont de plus en plus utilisées de façon systématique dans les méthodes de TAL. Le rôle des ontologies dépend alors de l'expressivité des langages utilisés pour les écrire. Pour les cas les plus simples, l'ontologie est un lexique qui permet de reconnaître les entités ou les concepts évoqués dans les textes [MKS04]. Pour les cas plus complexes, l'ontologie guide la reconnaissance de connaissances structurées dans les textes en fournissant un modèle des connaissances en question [RKK+00].

3.4.3 La recherche et l'interrogation de données

Le challenge relevé par les ontologies est la recherche et l'interrogation de façon homogène de diverses sources de données au sein desquelles les entités biologiques, par exemple une association à une maladie ou une implication dans un processus, sont nommées de façon différentes dans les sources. En effet, en biologie de nombreux synonymes, acronymes, abréviations, peuvent faire référence à une même entité. Un premier exemple est les différents noms donnés au processus de fabrication du glucose dans un organisme : (en gardant les termes anglo-saxons utilisés dans les sources)"glucose synthesis", "glucose biosynthesis", "glucose formation", "glucose anabolism", et "glucogenesis". Un second exemple concerne la présence d'un variant génétique à la position 2377 du gène *TMPT i.e.* le fait que le nucléotide à cette position du génome puisse être différent pour deux individus. Ce variant est identifié par "rs1142345" dans la base de données dbSNP, "TPMT*3C" dans la base OMIM, "Chr6 :18238897 A/G" dans PharmGKB, et "NC_00006.10g :18238897A>G" dans certaines publications scientifiques. Une ontologie peut proposer un identifiant unique sous la forme d'un terme ou de l'identifiant d'un concept pour chaque entité et peut également lui associer l'ensemble de dénominations alternatives. Dans ce cas

[40] http://obofoundry.org/
[41] http://www.bioontology.org/tools/portal/bioportal.html
[42] http://www.bioontology.org/tools/alpha.html

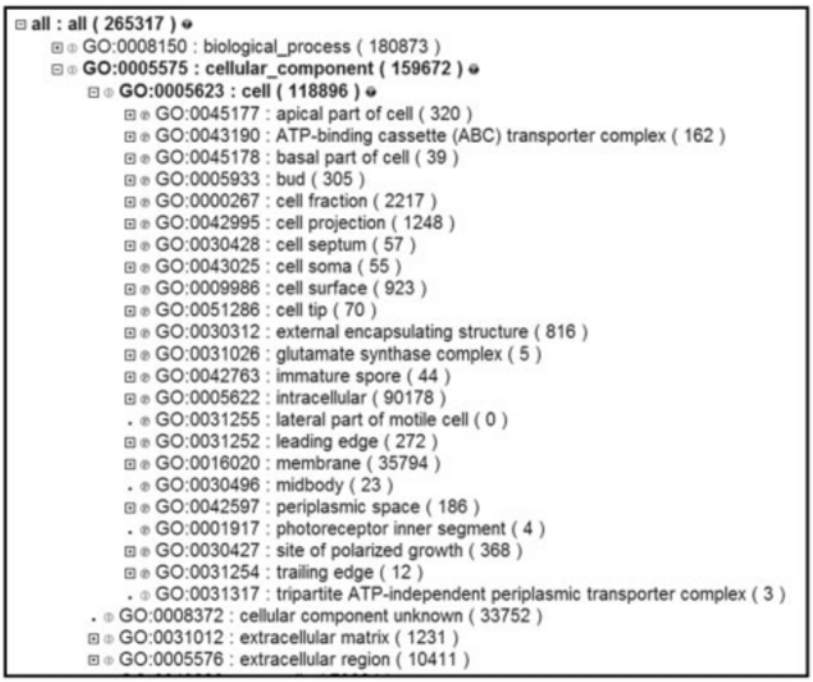

FIG. 2.9 – Extrait de la GENE ONTOLOGY

l'ontologie peut se présenter comme un *vocabulaire contrôlé* recouvrant un certain domaine et auquel peuvent être associés des ensembles de synonymes, acronymes, et abréviations. Les identifiants utilisés dans l'ontologie permettent alors d'identifier de façon consensuelle les entités biologiques représentées dans différentes sources.

L'exemple le plus connu de vocabulaire contrôlé utilisé en biologie est la GENE ONTOLOGY (GO) [ABB+00]. Les entités biologiques pour lesquelles elle propose un terme unique sont organisées selon trois hiérarchies relatives respectivement aux fonctions moléculaires, aux composants cellulaires, et aux processus biologiques. Ces termes sont supposés qualifier les produits de gènes, et plus précisément, leur fonction, où ils l'exercent et comment. La Figure 2.9 donne un aperçu des termes constituant la hiérarchie des composants cellulaires. Le succès de la GO revient à sa large utilisation pour l'*annotation* de produits de gènes dans diverses bases de données. A partir de cette annotation, les termes GO sont utilisés pour interroger de façon transparente et homogène plusieurs sources de données mais aussi pour analyser les résultats d'expérimentations à haut-débit.

3.4.4 L'échange de données entre applications

La définition d'une ontologie peut avoir comme objectif de spécifier un format d'échange standard pour un domaine. Il s'agit alors pour l'ontologie de décrire précisément les données, leurs types, et leurs relations. Ce genre d'application est intéressant lorsque les langages de représentation du Web sémantique (OWL), faciles à publier et à échanger sur le Web, sont utilisés. Les ontologies construites bénéficient au final de l'ensemble des technologies du Web sémantique nouvellement développées et peuvent être utilisées de surcroît pour l'intégration et l'analyse de données.

MAG_OM [BB06] et *BioPAX* [BC08] sont deux standards, spécifiés sous forme d'une ontologie et définis pour l'échange de données. Le premier est un modèle qui décrit les conditions expérimentales et les résultats d'expérience d'hybridation moléculaire en transcriptomique. Il est utilisé pour le partage de ce type de résultats entre chercheurs mais également pour la communication entre bases de données. Le

second, *BioPAX* est un standard d'échange d'informations sur les réseaux biologiques. Aussi, il n'a pas été conçu comme une représentation des connaissances du domaine mais son développement en OWL laisse la possibilité de le considérer comme tel et de l'employer pour des usages autres que l'échange de données. Par exemple, le fait que des sources de données sur les réseaux biologiques (comme HumanCyC, Reactome) fournissent un accès à leur contenu dans le format *BioPAX* a permis l'émergence d'un système d'intégration de données relatif aux réseaux appelé *Pathway Commons*[43].

3.4.5 L'intégration de données

Les ontologies ont la capacité de fournir une représentation du contenu de plusieurs bases de données biologiques et des relations entre les contenus des différentes bases. Les mécanismes de raisonnement peuvent ensuite utiliser de telles représentations pour, face une requête posée, déterminer quelles sont les ressources utiles pour y répondre et en extraire les éléments de réponse.

L'un des premiers systèmes à utiliser une bio-ontologie pour l'intégration de données est RiboWeb [ABC+99]. L'ontologie utilisée dans RiboWeb, composée de quatre parties, représente dans le langage Ontolingua [FFR97] à la fois le domaine biologique considéré (le ribosome) et les méthodes d'analyse possibles sur les données relatives à ce domaine. L'ontologie est instanciée à partir de diverses ressources disponibles sur le Web et relatives au ribosome pour constituer la BC RiboWeb. Les avantages de l'utilisation d'une ontologie sont ici essentiellement de proposer une représentation des données non seulement intégrée mais également associée à une sémantique (de façon manuelle). L'originalité principale de RiboWeb réside dans son utilisation concomitante d'une représentation des entités biologiques mais également des méthodes et objectifs d'analyse de ces entités qui facilitent cette analyse et l'interprétation de ses résultats.

TAMBIS est un système prototypique d'intégration de données qui suit une approche médiateur [SBB+00]. TAMBIS inclut notamment :

- une ontologie appelée TAMBIS ontology,
- une BC appelée le "modèle conceptuel biologique" qui fournit à l'utilisateur les concepts nécessaires à la construction de requêtes,
- un ensemble de mappings entre les concepts du "modèle conceptuel biologique" et les schémas des sources intégrées, et
- une interface graphique à partir de laquelle l'utilisateur peut définir sa requête.

L'utilisation de TAMBIS suit le processus suivant. Dans un premier temps, l'utilisateur utilise l'interface graphique pour construire une requête. La requête correspond à la description d'un concept de haut niveau formé à partir de concepts du "modèle conceptuel biologique" que l'utilisateur trouve en naviguant dans la hiérarchie de l'ontologie, et de constructeurs d'une LD. La requête ci-dessous est un exemple construit avec TAMBIS dans laquelle les termes en gras sont des noms individus de la BC , ceux en italique sont des constructeurs de la LD proposée, celui entre guillemets est une valeur, et les autres sont des rôles de l'ontologie.

protein *which*
  isHomologousTo **protein** *which*
    hasName "protein name :lard"
  and
  functionsInProcess **apoptosis**

Cette requête correspond à la question "quelles sont les protéines qui sont homologues à la protéine *lard* et dont la fonction est impliquée dans le processus biologique d'apoptose".

[43] http://www.pathwaycommons.org/

Dans un second temps, le système analyse les concepts utilisés dans la requête pour déterminer les sources pertinentes à interroger, et construit ensuite un plan de requêtes individuelles (*i.e.* chacune sur une seule source) en fonction des caractéristiques de la source et des contraintes exprimées dans la requête. Dans un troisième temps, le système exécute les requêtes propres à chaque source et collecte les résultats pour les retourner à l'utilisateur. La collecte des résultats se fait par le biais de *wrappers* adaptés spécialement à chaque source considérée. L'avantage principal de TAMBIS est de permettre l'expression de requêtes complexes comparé à des systèmes standard comme SRS[44] [EA93] ou Entrez[45] [Bax06], et ce grâce à l'utilisation des constructeurs de LD . Son inconvénient majeur survient lorsque l'on souhaite la déployer à l'échelle du Web et étendre la liste des sources qu'elle intègre. En effet un tel rafraîchissement demande l'enrichissement de l'ontologie, de la BC, la création de nouveaux mappings, et de nouveaux wrappers. La liste des sources intégrées par TAMBIS est très réduite ce qui fait de cet inconvénient une limite majeure à son utilisation.

Depuis TAMBIS, de nombreux systèmes utilisant des ontologies pour l'intégration de données biologiques ont vu le jour. Nous citerons seulement SEMEDA [KPL03] et BIOGUIDESRS [CBBDF07] qui se distinguent notamment par le fait qu'ils sont opérationnels.

3.4.6 Les mécanismes de raisonnement

L'utilisation des mécanismes de raisonnement associés aux ontologies constitue une application particulièrement prometteuse mais encore peu répandue. Nous citerons ici deux travaux de Wolstencroft *et al.* [WMS+05, WLT+06]. Dans le premier, il est fait usage des mécanismes de *vérification de consistance* et de *classification de concepts* sur une ontologie pour assister la construction du schéma d'une nouvelle base de données relatives aux familles de protéines phosphatases. Le second, détaillé chapitre 4 section 2.5.2, utilise les mécanismes de *classification de concepts et d'instances* pour permettre la classification automatique de protéines phosphatases dans leur famille et sous-famille.

Dans le domaine biomédical, des mécanismes de raisonnement sont également utilisés pour déduire les conséquences physiologiques de l'endommagement d'artères coronaires [RDM05]. Dans ce travail, l'ontologie FMA (*Foundational Model of Anatomy*) [RMM+98] est utilisée comme base pour représenter en OWL :

- la relation entre chaque artère coronaire et la zone du tissu cardiaque qu'elle vascularise,
- l'arborescence des artères vascularisant le cœur, notamment par une relation de continuité associant les artères connectées,
- l'occlusion d'une artère,
- l'ischémie du tissu cardiaque.

Un service de raisonnement associé à l'ontologie permet d'inférer, par un mécanisme de classification, le type de conséquence sur les tissus cardiaques que peut avoir un endommagement des artères.

Le système Kasimir utilise quant à lui le *raisonnement à partir de cas* en plus des mécanismes de raisonnement associés aux LD [dLN07]. Ces mécanismes sont appliqués à des protocoles de soins du cancer du sein, représentés en OWL, pour l'aide au diagnostic.

3.5 Vers une intégration semi-automatique de sources

Les sections précédentes montrent que l'intégration de données, même lorsqu'elle s'appuie sur des ontologies, présente encore de nombreuses limites tout en posant dans l'industrie comme dans la recherche des défis cruciaux. Il est illusoire d'espérer voir à court terme l'émergence d'outils, par exemple basés sur les technologies du Web sémantique, capables d'intégrer de façon entièrement automatique des sources

[44] http://srs.ebi.ac.uk/
[45] http://www.ncbi.nlm.nih.gov/Entrez/

de données hétérogènes. Cependant, les avancées dans le domaine de l'alignement d'ontologie constituent des pistes qui peuvent tout au moins réduire l'intervention manuelle nécessaire à l'intégration de données. Ainsi An *et al.* décrivent un outil semi-automatique appelé MAPONTO qui permet la mise en correspondance entre des schémas de bases de données (relationnelles ou XML) et une ontologie [AMB06]. Le travail de Leser et Naumann [LN05] constitue un proposition comparable appliquée à l'intégration de bases de données biologiques. Une direction intéressante évoquée par Euzenat *et al.* pour la définition semi-automatique de tels mappings est l'utilisation des capacités de raisonnement associées aux formalismes logiques [ES07].

La section 2 de ce chapitre présentait les ontologies comme un moyen de représenter les connaissances d'un domaine. Une ontologie peut notamment être utilisée dans le cadre de l'intégration de données où elle peut jouer un rôle analogue à un schéma global comme évoqué dans cette section. Dans ce cas, les avantages à utiliser une ontologie sont multiples : celle-ci est associée à une sémantique clairement définie suivant laquelle il est possible de mettre en accord les schémas de sources hétérogènes ; elle permet l'utilisation de mécanismes de raisonnement capables de vérifier la consistance de l'ontologie ; pour une ontologie du Web sémantique, elle s'appuie sur un ensemble de technologies qui facilitent son partage et son développement.

La contribution présentée dans le chapitre 3 propose une utilisation d'ontologies originales pour l'intégration de données. La contribution présentée chapitre 4 réutilise les même ontologies, ainsi que le résultat de l'intégration pour guider l'extraction de connaissances. La section suivante (4) est un état de l'art de l'utilisation des ontologies pour guider l'extraction de connaissances.

4 Extraction de Connaissances *guidée par* les Connaissances du Domaine – ECCD

Différents auteurs, dont Anand [ABH95], Phillips [PB01], Gottgtroy [GKM04], Cespivova [CRS+04], Lieber [LNST08], et plus généralement les ateliers internationaux SWM [SHB01, BHS02], KDO [BFG+04, ABG+06], et PriCKL [BSc07] se sont intéressés à l'utilisation de connaissances du domaine formalisées dans des ontologies, pour guider l'analyste et les machines dans le processus d'extraction de connaissances.

C'est notamment sur cette idée générale qu'est fondé le processus d'*Extraction de Connaissances guidée par les Connaissances du Domaine* (ECCD ou KDDK pour *Knowledge Discovery guided by Domain Knowledge* en anglais) décrit par Lieber *et al.* [LNST08]. Dans l'ECCD, les unités de connaissances extraites et validées sont exprimées dans un formalisme de représentation des connaissances afin d'être intégrées à une ontologie du domaine. L'ontologie ainsi enrichie est alors réutilisée lors des itérations suivantes du processus. Lors de chaque itération du processus, chacune des étapes peut bénéficier d'abord des connaissances initiales, et ensuite des connaissances nouvellement acquises.

(i) Lors de l'étape de préparation des données, les connaissances facilitent l'intégration de données hétérogènes et aident à la sélection de sous-ensembles de données plus pertinents à fouiller.

(ii) Lors de l'étape de fouille de données, les connaissances permettent de spécifier des contraintes pour par exemple circonscrire, ou au contraire élargir, l'espace de recherche des algorithmes.

(iii) Lors de l'étape d'interprétation des unités extraites, les connaissances aident à la visualisation et la validation des résultats.

L'ontologie de domaine est associée en permanence à des mécanismes de raisonnement capables de produire des règles d'inférence potentiellement utiles. Suivant ce cadre général décrit par l'ECCD , différents travaux se sont appliqués à étudier comment, en pratique, l'extraction de connaissances pouvait tirer parti de connaissances formalisées plus ou moins précisément. Les sections suivantes résument ceux qui nous ont paru les plus intéressants, que ce soit lors de la préparation, de la fouille ou de l'interprétation.

4.1 Préparation de données guidée par les connaissances

Il est ici question de l'utilisation de connaissances formalisées dans des ontologies pour assister les tâches d'intégration, de nettoyage, de transformation et de réduction de données présentées chapitre 2, section 1.

Intégration. L'utilisation d'ontologies lors de l'extraction et l'intégration de données, largement étudiée, a été abordée dans la section 3 de ce chapitre.

Nettoyage. Perez-Rey *et al.* ont développé l'outil *OntoDataClean* qui utilise l'ontologie *OntoDataClean preprocessing ontology* représentée Figure 2.10, pour aider au cours de l'étape de nettoyage des données à résoudre les problèmes d'inconsistance ou de données manquantes [PRAC06]. Pour utiliser cet outil, l'analyste doit décrire dans l'ontologie (*i.e.* en instanciant les concepts et rôles) l'enchaînement des opérations qu'il souhaite appliquer aux diverses bases de données considérées. Ces opérations sont par exemple le remplacement des valeurs manquantes ou la suppression de tuples trop bruités. Le système est ensuite capable, en se référant aux opérations décrites dans l'ontologie, de nettoyer les données de façon automatique. L'ontologie est ainsi utilisée pour aider l'analyste à comprendre les différentes opérations possibles lors de cette étape et à garder une trace des différentes stratégies adoptées.

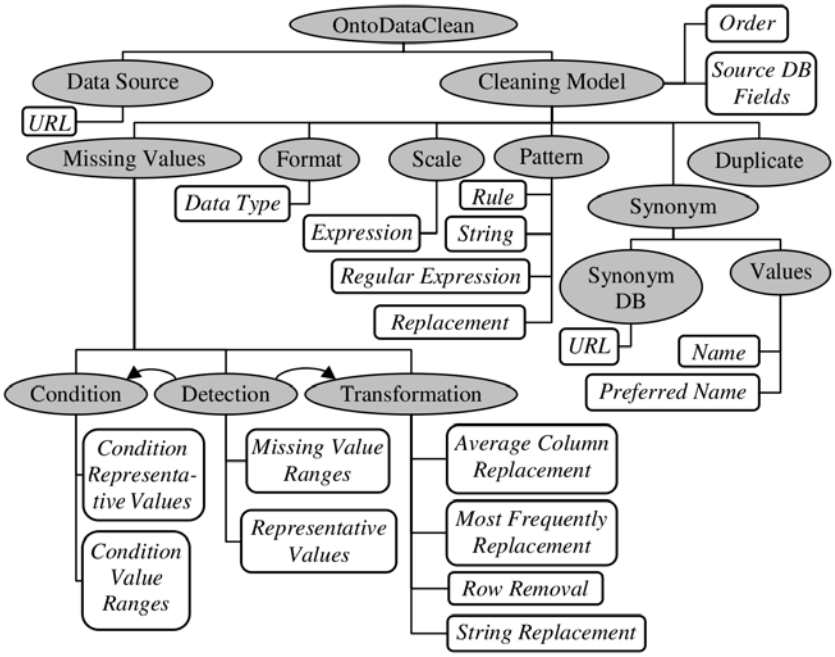

FIG. 2.10 – L'ontologie *OntoDataClean preprocessing ontology* présentée par Perez-Rey *et al.* [PRAC06]. Les ellipses grisées sont les concepts et les rectangles blancs leurs instances. Les lignes simples sont des relations de subsomption ou des assertions de concepts. Les lignes fléchées sont les rôles.

Transformation. Dans le cadre du projet *MiningMart*, Euler et Scholz proposent un outil d'aide à la transformation de données qui utilise deux ontologies. La première ontologie doit être manuellement construite en fonction du domaine étudié. Son but est double : (a) fournir un modèle plus proche de la conceptualisation du domaine de l'analyste et (b) fournir un modèle du domaine mettant en valeur les dépendances ou relations qu'il explore. La seconde ontologie doit permettre de décrire les opérations nécessaires et leur enchaînement afin de rendre possible la transformation des données originales dans un modèle qui correspond à la première ontologie (de domaine) spécialement construite. L'utilisation de cette ontologie a un rôle similaire à celui de Perez-Rey *et al.* : l'appréhension des opérations possibles et la documentation des transformations effectuées.

Bernstein *et al.* utilisent une ontologie qui représente les méthodes de préparation et de fouille de données pour aider l'analyste à définir une stratégie d'ECBD optimale [BPH05]. Pour cela l'analyste décrit la stratégie d'ECBD de son choix (objectifs, jeux de données, etc.) avec les concepts de l'ontologie. Le système appelé *Intelligent Discovery Assistant* relié à l'ontologie propose une liste d'enchaînement de méthodes de préparation et de fouille compatibles entre eux et avec le format initial des données, puis fournit un classement des enchaînements proposés selon des critères objectifs comme le temps d'exécution ou la précision des résultats.

Réduction. Liu *et al.* [LY05] ainsi que Guyon *et al.* [GE03] suggèrent d'utiliser les connaissances du domaine en première intension pour réduire le jeu de données en éliminant les attributs qui ne sont pas pertinents. En pratique de nombreuses approches d'ECBD utilisent de façon implicite les connaissances de l'analyste lors d'une sélection manuelle d'attributs d'intérêt. Cheng *et al.* [CWT06] comparent de façon empirique les méthodes automatiques de sélection (présentées en section 1) à une méthode faisant

intervenir l'expertise de l'analyste. Dans le cadre de cette étude, la seconde méthode montre une amélioration de la sensibilité de la classification proposée. Cependant cette préparation manuelle se limite aux connaissances propres de l'analyste sans se référer aux connaissances potentiellement disponibles par ailleurs. Certaines études suggèrent l'utilisation de connaissances mises à disposition de l'expert sous forme de listes d'attributs d'intérêt ou de listes de phrases pour améliorer les résultats de méthode d'ECBD ou de fouille de texte [Gai89, AFC99, CFCH01]. Dans ces cas la sélection (ou l'annotation d'un document) reste manuelle, mais l'analyste se réfère à une source de données extérieure qu'il interprète en terme de connaissances.

Wilcox *et al.* et Yu *et al.* ont proposé d'utiliser des connaissances représentées sous forme de contraintes ou de règles de telle sorte à ce que le système puisse aider à la sélection de données [WH03, YSS07]. Wilcox *et al.* ont montré, dans le cadre de leur étude de classification de documents, que l'utilisation de connaissances est un critère plus important que le choix de la méthode de classification employée (pour un ensemble de méthodes sélectionnées pour être traditionnellement utilisées pour cette tâche). Le travail de Yu *et al.* présente la particularité de coupler une méthode de sélection automatique, basée sur une méthode à noyau, et l'utilisation de connaissances. Les principales critiques qui peuvent être faites à ces deux travaux sont l'absence d'une sémantique associée aux connaissances considérées, la faible implication de l'analyste dans le processus de sélection, et la propension à sélectionner des attributs qui permettront la découverte de connaissances plus *attendues* que *nouvelles et surprenantes*.

Un travail récent propose l'utilisation d'une ontologie en LD pour guider la sélection d'attributs dans le cadre d'une opération appelée le *design de tâche* (*task design*) [SRR05]. L'ontologie sert alors à partitionner l'ensemble des attributs en différentes classes, par le biais d'un *mapping données-ontologie*, et permet ainsi à la fouille de travailler sur des partitions plus homogènes et donc plus riches en régularités. Cette méthode semble particulièrement pertinente puisqu'elle combine connaissances d'une ontologie et orientation de la sélection selon l'objectif de l'analyste. Toutefois la méthode proposée reste très générale et sa mise en œuvre contraignante puisque la mise en correspondance des données à fouiller et des concepts de l'ontologie dépend du domaine et est donc réalisée de manière *ad hoc*) et que la description des partitions potentiellement porteuses de régularités est manuelle.

Hormis celles qui concernent l'intégration de données, les méthodes faisant usage d'ontologies pour guider la préparation des données sont finalement peu répandues. La préparation est pourtant une phase déterminante pour la suite du processus durant laquelle l'analyste est particulièrement sollicité. C'est particulièrement le cas lors de la sélection de données, étape cruciale lorsque les méthodes de fouilles génèrent des résultats volumineux. La sélection de données est justement une tâche où les connaissances du domaine sont particulièrement utiles ce qui justifie leur utilisation de façon semi-automatique lorsqu'elles sont formalisées dans une BC relative au domaine étudié.

Dans la section 1 du chapitre 4, nous proposons pour guider la sélection d'utiliser une BC basée sur une ontologie de domaine et instanciée à partir du contenu des bases de données relatives. De cette façon l'analyste peut sélectionner un jeu de données à fouiller en prenant en compte ses propres connaissances, celles formalisées dans la BC et bénéficier des mécanismes de raisonnement associés (subsomption, classification).

4.2 Fouille de données guidée par les connaissances

Faire usage de connaissances formalisées au moment de l'étape centrale de fouille est délicat puisque cela nécessite la conception ou la modification d'un algorithme de fouille de sorte que celui-ci prenne en considération des éléments de connaissance. Nazeri et Bloedorn présentent dans [NB04] des modifications des algorithmes `Apriori` et `C4.5` qui visent à produire des RA en prenant en compte des éléments de connaissance du domaine. Les éléments de connaissance sont dans ce cas des listes de règles (que

nous appellerons aussi BC) représentées selon un formalisme défini précisément et non associé à une sémantique. Dans la version originale d'`Apriori` le seul critère d'inclusion d'un motif[46] est son support. Dans la version modifiée proposée, c'est d'abord la classe à laquelle le motif appartient dans la BC lorsqu'il y est représenté qui est déterminante. Ainsi,

- si le motif est dans la BC et appartient à la classe "motifs intéressants" alors il est conservé pour produire les RA quelque soit son support ;
- inversement, si le motif appartient à la classe "motifs inintéressants" alors il est éliminé quelque soit son support.

L'algorithme 2.1 représente simplement l'algorithme `Apriori` et les modifications (en gras) proposées par Nazeri et Bloedorn. Dans le cadre d'expérimentations menées par les auteurs avec `Apriori` modifié, le nombre de règles inintéressantes diminue sans que ne soient perdues les règles intéressantes par rapport à l'utilisation d'`Apriori` classique.

Algorithme 2.1

```
générer les motifs fréquents de longueur 1
Pour (n=2 à max) faire :
   générer les motifs candidats de longueur n
   Pour (chaque nouveau candidat) faire :
      vérifier si le motif est dans la BC
      Si (motif ∈ "motifs inintéressants")
         éliminer le motif
      Sinon si (motif ∈ "motifs intéressants")
         conserver le motif
      Sinon
         Si (support(motif) ⩾ support_min)
            conserver le motif
générer les RA
```

Dans le cas de l'algorithme `C4.5` qui est une méthode de construction d'arbre de décision, le choix de l'ordre des attributs qui permet la construction de l'arbre de décision est modifié de telle sorte que des attributs ayant un score faible soient choisis en priorité. En effet, la BC associe à certains attributs un score utilisé pour pondérer le classement initial des attributs et ainsi l'ordre des attributs choisis pour construire l'arbre. De nouvelles associations sont déduites de l'arbre puis utilisées pour modifier les scores associés aux attributs dans la BC afin d'être pris en compte lors des exécutions suivantes.

Karel et Kléma [KK07] proposent également de contraindre un algorithme de fouille (de recherche de RA quantitatives) en réduisant l'espace de recherche. Le jeu de données exploré, relatif à la transcriptomique, est composé d'attributs correspondant à des niveaux d'expression de gènes. Les éléments de connaissance sur lesquels s'appuyent les auteurs sont les hiérarchies de termes de la *Gene Ontology* (GO). Les termes de GO annotent (*i.e.* caractérisent) les gènes dans des bases de données, de la même façon que les niveaux d'expression viennent caractériser ces même gènes. Ceci permet d'associer les annotations GO et les attributs relatifs au niveau d'expression de gènes. La hiérarchie de l'ontologie sert alors à définir une mesure de similarité entre gènes qui représente le fait qu'il est plus ou moins "plausible" qu'un couple de gène soit co-exprimé. Lorsque les RA sont produites, celles qui proposent d'associer des gènes dont la co-expression est plausible sont préférées aux autres.

[46] Les motifs sont les éléments de base pour la génération des RA avec l'algorithme `Apriori` .

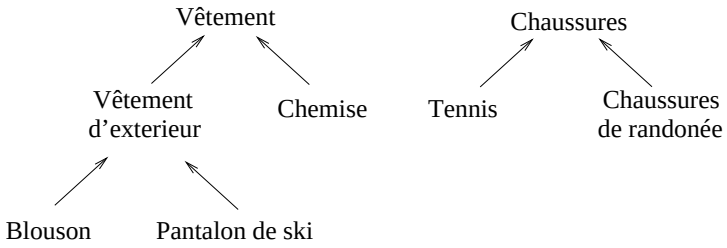

FIG. 2.11 – Taxonomie $\mathcal{T}$

Transaction	Produits achetés
100	Chemise
200	Blouson, Chaussures de randonnée
300	Pantalon de ski, Chaussure de randonnée
400	Chaussures
500	Chaussures
600	Blouson

TAB. 2.6 – Base de données $\mathcal{D}$

4.3 Interprétation guidée par les connaissances

Les méthodes de fouille sont susceptibles de produire des quantités de résultats importantes qui rendent la tâche d'interprétation fastidieuse pour l'analyste. C'est notamment le cas de la recherche de règles d'association (RA) qui produit des règles à la fois nombreuses et redondantes. Pour résoudre ce problème d'analyse des RA, de nombreuses mesures d'intérêt *objectives* et *subjectives* ont été proposées pour permettre le classement des règles [TKS02, McG05, Bri06]. L'intérêt d'une règle est un paramètre en partie subjectif, lié aux attentes de l'analyste, à ses propres connaissances mais aussi potentiellement lié aux connaissances du domaine disponibles. Une taxonomie peut ainsi être utilisée pour l'analyse des RA et la généralisation des règles [SA95]. Suivant cette méthode, un ensemble de règles $R = \cup(P_i \rightarrow C_i)$ dont l'ensemble des prémisses $\cup P_i$ sont fils d'une même classe P_p de la taxonomie et dont l'ensemble des conclusions $\cup C_i$ sont également fils d'une même classe C_p, ces règles peuvent être généralisées en une seule règle de forme $P_p \rightarrow C_p$. Par exemple, le Tableau 2.6 représente une base de données $\mathcal{D}$ de transactions de magasin et la Figure 2.11 une taxonomie des produits du magasin. Avec un support de 0,3 (*i.e.* 2 transactions) et une confiance de 0,6, les quatre règles obtenues en utilisant la généralisation sont représentées dans le Tableau 2.7. Les règles <Pantalon de ski ⇒ Chaussure de randonnée> et <Blouson ⇒ Chaussures de randonnée> ne satisfont pas les support et confiance minimums (respectivement $\frac{1}{6}$ et $\frac{1}{6}$), ce qui en revanche est le cas de la règle plus générale <Vêtement d'extérieur ⇒ Chaussures de randonnée> (support = $\frac{2}{6}$).

Règle	Support	Confiance
Vêtement d'extérieur ⇒ Chaussures de randonnée	0,33	0,66
Vêtement d'extérieur ⇒ Chaussures	0,33	0,66
Chaussures de randonnée ⇒ Vêtement d'extérieur	0,33	1
Chaussures de randonnée ⇒ Vêtement	0,33	1

TAB. 2.7 – Règles conservées ($\texttt{support}_{min}$=0,3, $\texttt{confiance}_{min}$=0,6) après généralisation

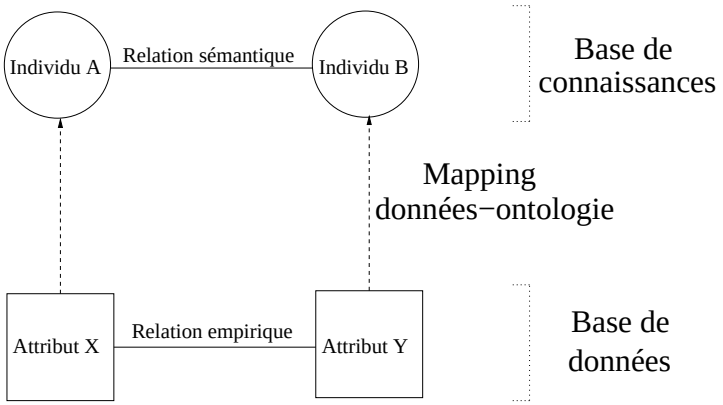

FIG. 2.12 – Mapping simple proposé dans [SRR05] pour guider l'interprétation des résultats de fouille

En plus d'une taxonomie, Liu *et al.* [LHCM00] utilisent un formalisme particulier pour représenter des modèles de règles que l'analyste s'attend à découvrir. Le modèle général d'une règle est de la forme

$$<P_1, P_2, ..., P_n \Rightarrow C_1, C_2, ..., C_n> [support][confiance]$$

où les éléments de $P_1, P_2, ..., P_n$ et $C_1, C_2, ..., C_n$ sont soit un attribut (et un terme de la taxonomie), soit un motif, soit une expression régulière pour décrire une classe de motifs. Le système associé mesure une distance entre chaque règle trouvée et les modèles proposés de façon à chiffrer le caractère *inattendu* des règles trouvées. Les règles les plus différentes des modèles de règles proposés par l'analyste sont les plus inattendues. Par exemple un modèle défini comme suit

$$<\{\text{Chaussures de randonnée, Chaussure}\}+ \Rightarrow \text{Chemise}+ >$$

signifie que les règles associant au moins un des attributs Chaussures de randonnée ou Chaussure avec au moins l'attribut Chemise sont attendues. Ceci permet par exemple de mettre en avant le caractère inattendu de la règle

$$<\text{Chaussures de randonnée} \Rightarrow \text{Vêtement d'extérieur}>$$

dont la conclusion ne contient pas l'attribut Chemise.

Un autre moyen de faciliter l'interprétation est le développement d'outils de visualisation et de validation des résultats. Svatek *et al.* [SRR05] ou Vanzin *et al.* [VB05] proposent des systèmes fondés sur le même principe d'un mapping données-ontologie préalablement établi. Celui-ci permet d'exploiter les relations de l'ontologie et la sémantique associée pour aider l'analyste à interpréter les relations empiriques mises en évidence lors de la fouille (voir Figure 2.12).

L'outil de visualisation inclus dans le système permet d'évaluer et d'interpréter les résultats de fouille en affichant et en permettant la navigation au travers des relations de l'ontologie associées aux résultats.

Les approches présentées dans cette section montrent la nécessité de définir un mapping entre les données analysées et l'ontologie. La définition de tels mappings a été abordée dans la section concernant l'utilisation des ontologies pour guider l'intégration de données (section 3.3). Dans le cas des travaux relatifs à l'extraction de connaissances nous observons qu'il s'agit le plus souvent de méthodes heuristiques et qu'aucune approche générale n'est proposée pour la définition ou la formalisation de tels mappings. De plus, la difficulté à faire correspondre des valeurs (les données) et des objets (les instances de l'ontologie) au sein des mappings n'est pas abordée dans ces travaux. Cela nous conduit à proposer d'exploiter

les résultats des travaux sur l'intégration sémantique pour développer des stratégies d'utilisation des connaissances dans un processus d'ECBD (Chapitre 4).

Par ailleurs il est possible de distinguer deux types d'exploitation de la sémantique associées aux ontologies selon le type d'ontologie considérée. D'une part les ontologies qui ne sont pas associées à une sémantique précise comme les taxonomies ou les vocabulaires contrôlés présentent l'avantage d'être faciles à manipuler et ainsi de tirer parti au maximum du peu de sémantique qui leur est associé. Par exemple, elle peuvent être facilement associées au contenu de bases de données ou de pages Web qu'il est alors possible d'analyser en considérant la structure de l'ontologie comme un lien entre tuples ou pages Web. C'est notamment le cas pour les bases de données biologiques annotées avec la GO et le travail de Karel et Klema [KK07]. D'autre part, les travaux usant d'ontologies représentées selon un formalisme associé à une sémantique précise, comme les LD, font un usage minimal de cette sémantique malgré les contraintes imposées par leur exploitation. La capacité naturelle des objets à représenter un domaine (ainsi à mieux le comprendre et à le faire comprendre) et l'organisation hiérarchique demeurent les deux principales propriétés utilisées pour faciliter l'extraction de connaissances. En revanche, les possibilités offertes par l'expressivité des formalismes utilisés et par les mécanismes de raisonnement sont quant à elles plus rarement utilisées.

Nous pensons que le développement des technologies du Web sémantique est une opportunité qui offre la possibilité de tirer le meilleur parti d'une sémantique formelle et des mécanismes de raisonnement associés. L'objectif de cette thèse est notamment d'exploiter au maximum ces possibilités pour guider la découverte de connaissances en biologie.

Chapitre 3

Ontologies pour l'intégration de données en pharmacogénomique

Ce chapitre présente le construction de deux ontologies originales SNP-Ontology et SO-Pharm, et leur utilisation pour l'intégration de données pharmacogénomiques. La particularité principale de l'approche utilisée pour l'intégration est de transformer les résultats de requêtes pour peupler une Base de Connaissance (BC) qui servira, par la suite, à guider l'extraction de connaissances (voir chapitre 4).

La section 1 de ce chapitre décrit tout d'abord la méthodologie rigoureuse adoptée pour construire nos ontologies, puis détaille chacune des étapes de cette méthodologie mises en œuvre dans le cas de la construction de l'ontologie SNP-Ontology puis de l'ontologie SO-Pharm. La section 2 propose une méthode d'intégration de données qui utilise les ontologies d'une façon similaire à un schéma global dans une approche d'intégration de type médiateur. Les sections 3.1 et 3.2 décrivent les applications de cette méthode et les expérimentations conduites avec des données relatives aux variations génomiques et pharmacogénomiques. Enfin, la section 4 discute les résultats obtenus.

1 Construction d'ontologie : méthodologie proposée et mise en œuvre

Avant de pouvoir utiliser une ontologie, il est évidemment nécessaire de la construire. Une telle construction est un travail long et délicat qui demande une collaboration entre ingénieurs des connaissances, maîtrisant les méthodes de représentation des connaissances et experts du domaine, maîtrisant les connaissances à représenter. Afin de valoriser les efforts engagés lors de leur construction, des ontologies existantes sont partagées dans des *bibliothèques d'ontologies* sur le Web, comme c'est par exemple le cas dans le domaine de la biologie avec les sites OBO Foundry[47] et BioPortal[48]. La mise à disposition de ces ontologies peut en théorie éviter la reconstruction de nouvelles ontologies pour les domaines déjà couverts. Cependant, la conceptualisation d'une ontologie dépend étroitement de la définition du domaine qu'elle représente et des objectifs liés à sa construction, c'est pourquoi il est rare, en pratique, qu'une ontologie existante convienne en même temps au domaine et aux objectifs d'un nouveau travail. Dans un premier cas extrême, aucune ontologie ne correspond au domaine et objectifs, il est alors nécessaire de construire entièrement une nouvelle ontologie. Dans un deuxième cas plus courant, les ontologies existantes couvrent partiellement le domaine et répondent partiellement aux exigences imposées par les objectifs. Une démarche rationnelle consiste alors à réutiliser les ontologies existantes en les adaptant à ses propres domaine et objectifs.

[47] http://obofoundry.org/

[48] http://www.bioontology.org/tools/portal/bioportal.html

Cette section présente d'abord une méthodologie de construction d'ontologie inspirée des méthodes décrites dans la littérature mais adaptée à notre propos. Une des particularités de cette méthodologie est d'inclure une étape de formalisation des relations éventuelles avec d'autres ontologies existantes est formellement décrite avant leur implémentation. Nous présentons ensuite (sections 1.2 et 1.3) les particularités associées à la mise en œuvre de cette méthode lors de la construction de deux ontologies : *SNP-Ontology*, qui représente des connaissances relatives aux variations génomiques (ou variants) et *SO-Pharm* qui englobe plus généralement le domaine de la pharmacogénomique.

1.1 Méthodologie de construction manuelle d'ontologies pour l'intégration de données

Des méthodes semi-automatiques comme la classification, la fouille de textes, peuvent être utilisées pour construire une ontologie [Ome01, BCM05]. Ces méthodes sont intéressantes pour constituer une représentation des connaissances à partir de schémas de bases de données ou de corpus de textes. En revanche, elles sont peu compatibles avec l'objectif principal de nos ontologies qui est de proposer une représentation des connaissances qui soit la plus proche possible des connaissances de l'expert et le plus indépendante possible de la structures des bases de données existantes avec l'idée que ceci facilite l'*intégration de données*, et l'*Extraction de Connaissances à partir de Bases de Données* (ECBD).

Les ontologies construites par des méthodes semi-automatiques proposent une représentation des connaissances marquée par la structuration et le format des sources de données qu'elles exploitent. Inversement, nous souhaitons une représentation la plus neutre possible vis à vis des sources, de manière à laisser possible la mise en correspondance de l'ontologie obtenue avec le contenu d'un maximum de sources hétérogènes existantes ou à venir. De plus, la construction semi-automatique d'ontologie est un champ de recherche à part entière. Les méthodes qui en émergent sont souvent dépendantes d'un domaine et d'un format de source et leur utilisation nécessite, en conséquence, une adaptation et une évaluation coûteuses en temps qui sortent du cadre de nos travaux. Pour ces différentes raisons, nous préférons une construction manuelle suivant une méthodologie définie de façon rigoureuse (décrite ci-après) et impliquant des experts du domaine.

La méthodologie adoptée correspond à l'adaptation à notre contexte des processus de construction *itératifs* décrits classiquement [UK95, FGPJ97, NM01]. De cette méthodologie ressortent cinq étapes : la *spécification*, la *conceptualisation*, la *formalisation*, l'*implémentation*, et enfin l'*évaluation*, dont les résultats conduisent à une nouvelle itération.

1.1.1 Spécification

Le *domaine* couvert par l'ontologie doit être clairement défini avec les experts. Cette définition inclut la précision des limites du domaine, éventuellement de ce que ne couvre pas l'ontologie, et du niveau de *granularité* demandé pour représenter les connaissances du domaine. Les *objectifs* pour lesquels l'ontologie est construite doivent aussi être précisément déterminés avec les experts.

Durant cette étape il est important de définir les *critères d'évaluation* selon lesquels l'ontologie sera jugée à la fin de chaque itération du processus de construction. Dans notre cas ces critères sont *(1)* la *consistance*[49] de l'ontologie, *(2)* la capacité à répondre aux *questions de compétence*, *i.e.* une liste de questions auxquelles l'ontologie doit permettre de répondre, *(3)* la capacité à *représenter explicitement* des connaissances implicites contenues dans des bases de données ou des publications scientifiques.

Des *règles de nommage* (début du nom avec ou sans majuscule, sans espace, liste des caractères acceptés, etc.) sont adoptées pour les noms de concepts, de relations entre concepts, d'individus, et les valeurs qui seront utilisés lors de la construction.

[49] Une ontologie est consistante si tous ces concepts peuvent être instanciés.

Ensuite, deux listes sont établies en parallèle : une *liste des termes du domaine* établie par l'expert et une *liste des sources de données et de connaissances* relatives au domaine. Les sources contenues dans la seconde liste peuvent être de nature très différente comme un modèle conceptuel (en UML ou en un langage apparenté), un schéma XML, une base de données, une ontologie OWL, ou encore un vocabulaire contrôlé. Des exemples concrets de telles listes de sources sont donnés dans la suite de ce chapitre. Les sources de cette liste sont par la suite explorées pour enrichir la liste initiale de termes.

Dans un deuxième temps, la liste des sources est utilisée pour identifier les sources de connaissances qui peuvent être réutilisées pour la construction de l'ontologie. Les sources de connaissances sont sélectionnées notamment en fonction de leur pertinence par rapport aux objectifs fixés, et en fonction de leur qualité. Les critères de qualité préconisés par l'initiative OBO Foundry[50] constituent une liste de critères sur lesquels il est possible de s'appuyer pour choisir les meilleures sources à réutiliser. Dans le cas où aucune source de connaissances n'est suffisamment pertinente pour être réutilisée dans la construction de l'ontologie alors l'ontologie doit être entièrement construite.

1.1.2 Conceptualisation

La conceptualisation du domaine se fait à l'aide de *diagrammes de classes* UML [RBJ00]. L'expressivité offerte par UML, l'adaptation des classes (*i.e.* de la représentation par objets) pour représenter les concepts, et l'ouverture du langage UML, font de ce type de diagramme un outil adapté à la conceptualisation d'une ontologie [KCH+02]. La liste de termes est utilisée pour identifier les concepts de l'ontologie, sous la forme de *classes UML*, auxquels sont assignés un nom et une définition précise sous la forme d'un texte libre. Une fois ces concepts identifiés, leurs relations hiérarchiques et non hiérarchiques sont modélisées sous forme d'associations dans les diagrammes de classes.

Les relations entre les *concepts propres* à la nouvelle ontologie et les *concepts externes* des ontologies réutilisées sont également définies durant la conceptualisation en diagramme de classes. Dans le cas présent, les relations proposées entre concepts propres et concepts externes sont restreintes à trois types particuliers de relations : la *généralisation*, l'*équivalence*, et l'*agrégation*.

Généralement le choix du type de relation entre deux concepts est déterminé par les experts qui prennent en considération leurs connaissances du domaine et les définitions des deux concepts. Cependant, dans certains cas, le choix du type de relation entre deux concepts provenant de deux bio-ontologies est orienté par le type des ontologies considérées. En effet, les ontologies utilisées dans le domaine biomédical peuvent être divisées en trois catégories principales [RKM+05] :

- les *méta-ontologies* qui décrivent des concepts et rôles indépendants du domaine qui servent de modèle ou de composant pour les ontologies plus spécifiques (par exemple DOLCE[51], SUMO[52]) ;
- les *ontologies de domaines* qui représentent un certain domaine d'application et décrivent les entités qui lui sont relatives suivant un formalisme de représentation des connaissances (comme une Logique de Descriptions LD) ;
- les *vocabulaires contrôlés spécialisés* souvent développés manuellement par un consortium d'experts pour l'annotation des bases de données (par exemple GENE ONTOLOGY).

Typiquement une ontologie de domaine en LD va *généraliser* les concepts d'un vocabulaire spécialisé, c'est à dire que la description formelle d'un concept va généraliser un ensemble de concepts spécialisés. De façon similaire des ontologies dont le niveau d'abstraction est plus élevé, peuvent à leur tour *généraliser* les définitions des concepts de l'ontologie de domaine. Les ontologies que nous souhaitons construire sont des ontologies de domaine en LD qui proposent des relations vers des vocabulaires contrôlés. L'association de ces deux types d'ontologie permet de bénéficier conjointement de la sémantique

[50] le principes de qualité de l'OBO Foundry : `http://obofoundry.org/crit.shtml` (dernière visite le 17/07/2008)
[51] `http://www.loa-cnr.it/DOLCE`
[52] `http://www.ontologyportal.org/`

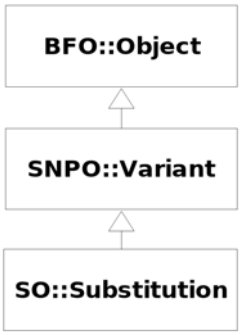

FIG. 3.1 – Extrait d'un diagramme de classes UML illustrant les relations de généralisation entre un concept issu d'un vocabulaire contrôlé, Sequence Ontology (SO), un concept d'une ontologie de domaine, SNP-Ontology (SNPO), et un concept d'une méta-ontologie, Basic Formal Ontology (BFO)

associée aux LD et de la richesse et de l'expertise associées aux vocabulaires contrôlés.

Par exemple, comme l'illustre la Figure 3.1, le concept propre de l'ontologie de domaine SNP-Ontology $SNPO :: variant$ généralise le concept externe $SO :: substitution$ et ses descendants issus du vocabulaire contrôlé Sequence Ontology. Par ailleurs, le même concept $SNPO :: variant$ est lui même généralisé par le concept externe $BFO :: Object$ importé d'une méta-ontologie et ainsi hérite et réutilise les définitions formelles du concept qui y est décrit.

1.1.3 Formalisation

La formalisation de l'ensemble de l'ontologie en LD ($\mathcal{SHOIN}(\mathcal{D})$) est menée de front avec son implémentation en OWL-DL, sauf pour ce qui concerne la formalisation des relations entre concept propre et concept externe (appartenant à une autre ontologie) qui est établie en LD au préalable. En fonction du type de relation choisi entre un concept propre et un concept externe lors de la conceptualisation, un axiome est décrit entre les concepts de LD correspondants notés C_{prop} et C_{ext} :

- la *généralisation* d'un concept propre par un concept externe est traduite par la relation de subsomption :
$$\mathsf{C}_{prop} \sqsubseteq \mathsf{C}_{ext} ,$$
- inversement, la *spécialisation* d'un concept propre par un concept externe est traduite par l'inverse de la subsomption :
$$\mathsf{C}_{prop} \sqsupseteq \mathsf{C}_{ext} ,$$
- l'*équivalence* entre deux concepts est formalisée par l'axiome :
$$\mathsf{C}_{prop} \equiv \mathsf{C}_{ext} ,$$
- la formalisation d'une relation d'*agrégation* entre deux concepts est notée :
$$\mathsf{C}_{prop} \sqsubseteq \exists \mathtt{isPartOf}.\mathsf{C}_{ext} ,$$
ou l'inverse si c'est le concept externe qui est une partie du concept propre.

L'exemple de relations entre concepts propres et externes représenté Figure 3.1 peut ainsi être noté comme suit :

$$\begin{array}{lcl} \text{SNPO :variant} & \sqsubseteq & \text{BFO :object} \\ \text{SNPO :variant} & \sqsupseteq & \text{SO :substitution} \end{array}$$

Des exemples concrets et plus variés des différents types d'axiomes possibles sont donnés dans la suite de ce chapitre.

Les domaines appelés en anglais *ontology matching*, *ontology alignment*, ou *ontology mapping*, s'intéressent au développement de systèmes d'alignement d'ontologies. Ces systèmes visent à établir, la plupart du temps de façon semi-automatique, des relations de généralisation ou d'équivalence entre les concepts de deux ontologies distinctes. Ils exploitent pour cela la similarité des noms de concepts, de leurs définitions formelles (mais aussi de leurs extensions, de leurs positions relatives dans une structure, etc.) pour proposer des relations entre concepts issus d'ontologies distinctes [ES07]. Nous privilégions ici la définition manuelle, par les experts du domaine, des relations entre concepts d'ontologies distinctes.

Des initiatives récentes, notamment le C-OWL [BGvH+03], clarifient la sémantique et enrichissent les types de relations possibles pour articuler des concepts d'ontologies distinctes.

1.1.4 Implémentation

La formalisation en LD et l'implémentation en OWL-DL sont imbriquées. Sur la base des diagrammes de classes, les concepts et leurs relations sont décrits formellement sous forme de concepts et rôles en LD à l'aide de l'éditeur d'ontologie Protégé [KFNM04].

Malheureusement, il n'existe pas de système automatique de conversion des diagrammes de classes UML en axiomes OWL. Aussi la conversion est faite manuellement. Les concepts et relations simples en UML sont directement traduits, en revanche les concepts plus complexes nécessitent une attention particulière. Par exemple, les LD ne permettent que la représentation de relations binaires. Cela rend relativement complexe la formalisation des relations n-aires représentées en UML. Le moyen le plus courant pour surmonter ce problème est appelé la *réification* [NR06]. Celle-ci permet d'éviter l'utilisation de relations n-aires lors de la conceptualisation, en préférant la construction de concepts supplémentaires, et notamment des concepts qui décrivent une relation n-aire et la décomposent en plusieurs relations binaires.

Pour être articulées avec la nouvelle ontologie, les ontologies précédemment sélectionnées doivent être implémentées dans le même langage, *i.e.* en OWL. Cela nécessite leur conversion lorsqu'elles ne sont pas directement disponibles dans ce langage. Elles sont ensuite importées et reliées à l'ontologie par l'implémentation en OWL des axiomes définis lors de l'étape précédente. Pour que l'implémentation de ces axiomes soit possible, il est nécessaire que les différentes ontologies articulées par les axiomes soient physiquement mises en présences. Il est alors nécessaire de spécifier le chemin d'accès et l'espace de nommage unique (*namespace* en anglais) des ontologies reliées de telle sorte que leurs concepts et rôles puissent être évoqués dans les descriptions de concepts propres à l'ontologie en construction.

D'un point de vue théorique, il est possible de considérer la liste d'axiomes entre concepts propres et externes comme une *TBox* à part entière. C'est notamment ce qui semble le plus pertinent du fait que ceci permet d'éviter d'avoir à incorporer des concepts externes dans la *TBox* d'une ontologie et de ainsi garantir l'intégrité de l'ontologie produite aussi bien que celle des ontologies articulées. Cependant, les contraintes qu'imposent la mise en œuvre d'une telle *modularisation* des ontologies limite son implémentation dans les outils standards d'édition d'ontologie tel que Protégé.

1.1.5 Évaluation

Elle se fait suivant les trois critères définis lors de la spécification : *consistance*, *questions de compétence*, et capacité à *représenter des connaissances* du domaine.

La *consistance* de l'ontologie et la classification de ses concepts sont vérifiées régulièrement au fur et à mesure et à l'issue de la formalisation/implémentation à l'aide des mécanismes de raisonnement

standards implémentés dans RacerPro [HM03].

La qualité des réponses aux *questions de compétences* est évaluée selon des critères définis lors de la spécification. Dans notre cas, les réponses à ces questions ne dépendent pas seulement de l'ontologie, mais également du système dans lequel elle est impliquée : un système d'intégration de données ou d'extraction de connaissances.

La capacité de l'ontologie à *représenter des connaissances* établies du domaine est évaluée par l'instanciation manuelle de l'ontologie à partir d'exemples de connaissances de deux origines différentes. Elles peuvent être soit extraites de bases de données, soit extraites de publications scientifiques du domaine.

L'évaluation de l'ontologie suivant l'ensemble de ces critères permet d'identifier des concepts et des rôles absents ou mal décrits dans l'ontologie. Ceux-ci sont alors pris en considération pour améliorer les spécification, conceptualisation et implémentation lors de l'itération suivante du processus de construction.

Il n'y a pas à proprement parler de critère d'arrêt de la construtions d'une ontologie. Certain auteurs utilisent, comme en génie logiciel, la notion de *cycle de vie* [DCGR98]. Un premier cycle de vie de l'ontologie se termine lorsque celle-ci est exploitée dans le cadre de l'utilisation pour laquelle elle a été développée. Cependant cette utilisation n'est pas forcement un aboutissement et peut donner lieu à l'identification d'imperfections qu'un nouveau cycle d'amélioration et d'enrichissement de l'ontologie visera à corriger.

1.2 Construction d'une ontologie pour les variations génomiques : SNP-Ontology

1.2.1 Spécification

Domaine couvert par SNP-Ontology L'objet de *SNP-Ontology* est de proposer une représentation formelle des variations génomiques. Ces variations génomiques sont des régions du génome clairement localisées dont la composition en nucléotides est susceptible de varier entre les individus d'une même espèce. La section 2 du chapitre 1 donne plus de détails sur les variations génomiques. La majorité de ces variations (environ 90% selon Kruglyak et Nickerson [KN01]) sont des variations ponctuelles, *i.e.* limitées à un nucléotide, alors appelées SNP pour *Single Nucleotide Polymorphism*. Malgré son nom, SNP-Ontology ne se limite pas à la représentation des SNP, mais représente les variations génomiques au sens large. Elle permet de représenter sans ambiguïté une variation génomique localisée sur une séquence d'ADN, ainsi que les conséquences que cette variation peut avoir au niveau du transcriptome (sur une séquence d'ARN) et du protéome (sur une séquence d'acides aminés). SNP-Ontology est développée de façon volontairement générale afin de permettre la représentation des variations du génome de différents organismes ainsi que les variations relativement à différentes versions d'un même génome. Une telle représentation n'était jusqu'alors pas disponible (tout au moins publiquement).

Les dernières versions de SNP-Ontology permettent de représenter les haplotypes et les variations du nombre de copies [RIF+06]. La représentation de notions complexes, comme l'influence d'une variation génomique sur l'épissage [HRT+05] ou encore sur la quantité de protéines traduites ne sont pas représentées mais constituent des pistes d'évolution pour ses versions futures.

Objectifs de SNP-Ontology La représentation non ambiguë des variations dans SNP-Ontology a pour objectif de permettre l'*intégration de données* hétérogènes relatives aux variations génomiques et à leurs conséquences. Pour cela, l'ontologie doit permettre *(1)* la représentation des variations suivant différents modes de description existants, *(2)* la représentation de l'équivalence entre deux descriptions distinctes d'une même variation, ainsi que *(3)* la correspondance entre une variation génomique et ses conséquences aux niveaux du transcriptome et du protéome. Par exemple, la variation notée *TPMT*3C* est équivalente à celle notée *Chr6 :18238897 A/G* et induit au niveau protéique une variation décrite par *TPMT :TYR240CYS*. L'objectif général de SNP-Ontology est de faciliter chaque étape du processus d'ECBD : préparation (y compris l'intégration), fouille, et interprétation.

Critères d'évaluation particuliers Des exemples de questions de compétence auxquelles SNP-Ontology doit répondre sont :

- Le gène humain *CYP2D6* présente-t-il des variations génomiques ?
- Si oui, certaines d'entre elles sont elles répertoriées à la fois dans les bases dbSNP et OMIM ?
- Certaines sont elles répertoriées dans la base PharmGKB et dans aucune autre ?
- Parmi ces mêmes variations lesquelles sont non-synonymes, *i.e.* localisées dans une région codante et qui entraîne une variation d'acides aminés dans la protéine résultante ?
- Certaines de ces variations sont elles localisées à une distance inférieure à 50 nucléotides en amont ou en aval des exons du gène *TPMT* ?
- Est-il possible de déterminer un ensemble de tag-SNP qui marquent les haplotypes auxquels appartiennent les variants de l'ensemble initial ?

SNP-Ontology doit permettre de représenter les connaissances qui peuvent être extraites des bases de données que l'on souhaite intégrer, *i.e.* les connaissances relatives aux variations génomiques enregistrées dans les bases dbSNP, OMIM, PharmGKB, HapMap, et dans des bases de données locus spécifiques.

Listes de termes et de sources de données et de connaissances relatives Une liste des termes utilisés dans le domaine et une liste des source de données et de connaissances relatives au domaine sont constituées. La liste des sources utilisées pour enrichir la liste des termes relatifs aux variations génomiques est présentée dans le Tableau 3.1. Seules deux sources de connaissances présentent un intérêt à être articulées avec SNP-Ontology : *AA Ontology* et *Sequence Ontology* dont une brève description est donnée Tableau 3.2.

Nom de la source	*Type de source*		*URL*
AA Ontology	Ontologie OWL	générique	http ://www.co-ode.org/ontologies/amino-acid/
dbSNP	schéma XML, modèle de données	générique	http ://www.ncbi.nlm.nih.gov/projects/SNP/
HapMap	schéma XML	humain	http ://www.hapmap.org/
HGVBase	DTD, modèle de données	humain	http ://hgvbase.cgb.ki.se/
BD inserm umrs538	DTD, modèle de données	humain, LS	privée
MECV	Vocabulaire contrôlé	générique	http ://www.ebi.ac.uk/mutations/
OMG SNP	Modèle de données	générique	http ://www.omg.org/technology/documents/formal/snp.htm
OMIM	Source de données	humain	http ://www.ncbi.nlm.nih.gov/omim/
PharmGKB	schéma XML, modèle de données	humain	http ://www.pharmgkb.org/
Sequence Ontology	Vocabulaire contrôlé	générique	http ://song.sourceforge.net/
LOVD	Source de données	humain, LS	http ://www.ucl.ac.uk/ldlr/LOVDv.1.1.0/
UMD LDLR	Source de données	humain, LS	http ://www.umd.be/LDLR/
Uniprot	Source de données	générique	http ://www.uniprot.org

TAB. 3.1 – Liste des sources explorées pour enrichir la liste de termes relatifs aux variations génomiques. La troisième colonne précise si la source de variations génomiques concerne uniquement un locus particulier (source Locus Spécifique ou LS), uniquement l'humain, ou si elle est générique (multi-locus et multi-espèces).

Ontologie	*Domaine*	*Prefixe*	*Namespace*
AA Ontology	acides aminés	AAO	http ://www.co-ode.org/ontologies/amino-acid/2005/10/11/amino-acid.owl
Sequence Ontology	Séquences et variations	SO	http ://purl.org/obo/owl/SO

TAB. 3.2 – Les deux ontologies articulées avec SNP-Ontology

1.2.2 Conceptualisation

La Figure 3.2 représente la répartition, sous forme de quatre paquets (ou *packages* en anglais), des diagrammes de classes correspondant à SNP-Ontology. Les Figures 3.3 et 3.4 sont deux exemples de diagrammes de classes centrés respectivement sur le concept de *variant* et sur celui de *séquence*. Ainsi la Figure 3.3 représente un variant comme un concept associé à une certaine position dans une séquence et associé à une variation observée (ObservedVariation) qui peut être soit une variation de nucléotide (NucleotideVariation) soit une variation d'acide aminés (AAVariation) selon le type de séquence sur laquelle le variant est observé. La Figure 3.4 représente notamment les séquences de nucléotide, leur composition en nucléotide, le fait qu'il peut s'agir soit d'une séquence d'ADN (DNASequence), soit d'une séquence d'ARN_m (mRNASequence) et entre autres que les séquence d'ADN compose les chromosomes et les gènes.

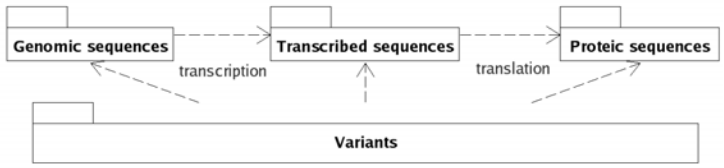

FIG. 3.2 – Diagramme UML représentant la répartition des diagrammes de classes en quatre paquets (*packages* en anglais). Le concept de variant peut être associé aux séquences génomiques sur lesquels ils sont localisés originellement, mais aussi aux séquences transcrites et protéiques sur lesquelles sont observées les conséquences des variations génomiques.

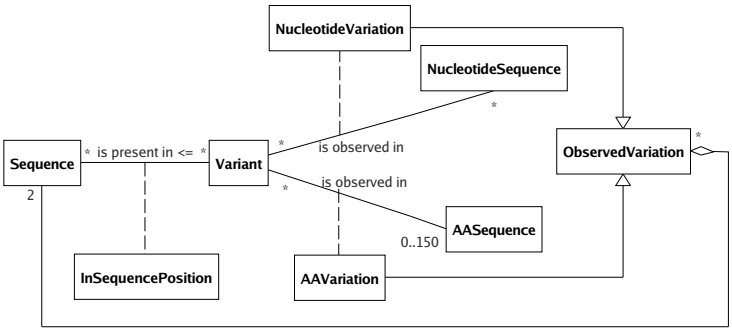

FIG. 3.3 – Diagramme de classes UML conceptualisant un variant, la variation observée pour un variant et sa position sur une séquence

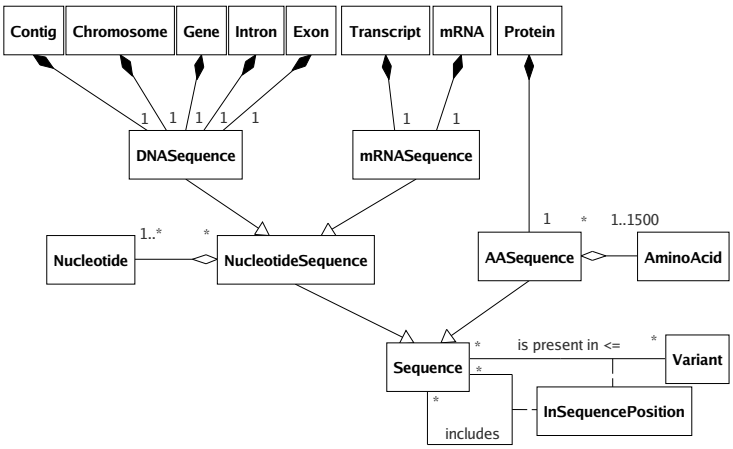

FIG. 3.4 – Diagramme de classes UML relatif aux séquences associées à un variant

1.2.3 Formalisation

Les relations décrites entre concepts propres à SNP-Ontology et concepts externes sont traduits en LD sous forme d'axiomes. Le Tableau 3.3 liste les axiomes reliant SNP-Ontology à l'AA Ontology et Sequence Ontology.

SNPO :amino_acid ≡ AAO :AminoAcid
SNPO :assembly ≡ SO :assembly (SO :0000353)
SNPO :contig ≡ SO :contig (SO :0000149)
SNPO :chromosome ≡ SO :chromosome (SO :0000340)
SNPO :codon ⊒ SO :transcription_start_site (SO :0000315)
SNPO :codon ⊒ SO :transcription_stop_site (SO :0000616)
SNPO :exon ⊒ SO :exon (SO :0000147)
SNPO :intron ⊒ SO :intron (SO :0000188)
SNPO :gene ≡ SO :gene (SO :0000704)
SNPO :genome ≡ SO :genome (SO :0001026)
SNPO :promotor ≡ SO :promotor (SO :0000167)
SNPO :terminator ≡ SO :terminator (SO :0000141)
SNPO :cnvr ≡ SO :copy_number_variation (SO :0001019)
SNPO :repeated_segment ⊒ SO :repeat_region (SO :0000657)
SNPO :haplotype ≡ SO :haplotype (SO :0001024)
SNPO :transcript_region ≡ SO :transcript_region (SO :0000833)
SNPO :mature_mrna ≡ SO :RNA (SO :0000356)
SNPO :transcript ≡ SO :transcript (SO :0000673)
SNPO :genomic_region ⊒ SO :QTL (SO :0000771)
SNPO :genomic_region ⊒ SO :pseudogenic_region (SO :0000462)
SNPO :genomic_region ⊒ SO :intergenic_region (SO :0000605)
SNPO :genomic_region ⊒ SO :regulatory_region (SO :0005836)
SNPO :genomic_region ⊒ SO :binding_site (SO :0000409)
SNPO :genomic_region ⊒ SO :haplotype_block (SO :0000355)
SNPO :genomic_region ⊒ SO :chromosome_part (SO :0000830)
SNPO :genomic_region ⊒ SO :regulatory_region (SO :0005836)

TAB. 3.3 – Liste des axiomes décrivant les relations entre concepts propres à SNP-Ontology (SNPO) et concepts externes importés de AA Ontology (AAO) et Sequence Ontology (SO). Les identifiants des concepts de SO sont donnés entre parenthèses.

1.2.4 Implémentation

Les Figures 3.5 et 3.6 schématisent certains concepts et rôles de SNP-Ontology. Ces deux figures peuvent être comparées aux diagrammes de classes UML (Figures 3.3 et 3.4) pour illustrer la conversion entre diagrammes de classes UML et LD. SNP-Ontology est disponible en OWL-DL sur le Web à l'adresse suivante : `http://www.loria.fr/~coulet/snpontology1.4_description.php`.

Sa version 1.4 contient 69 concepts, dont 21 concepts définis, et 59 rôles.

Concernant la conversion en OWL des ontologies articulées, AA Ontology est développée en OWL, donc elle ne nécessite aucune conversion. En revanche, Sequence Ontology est développée dans un for-

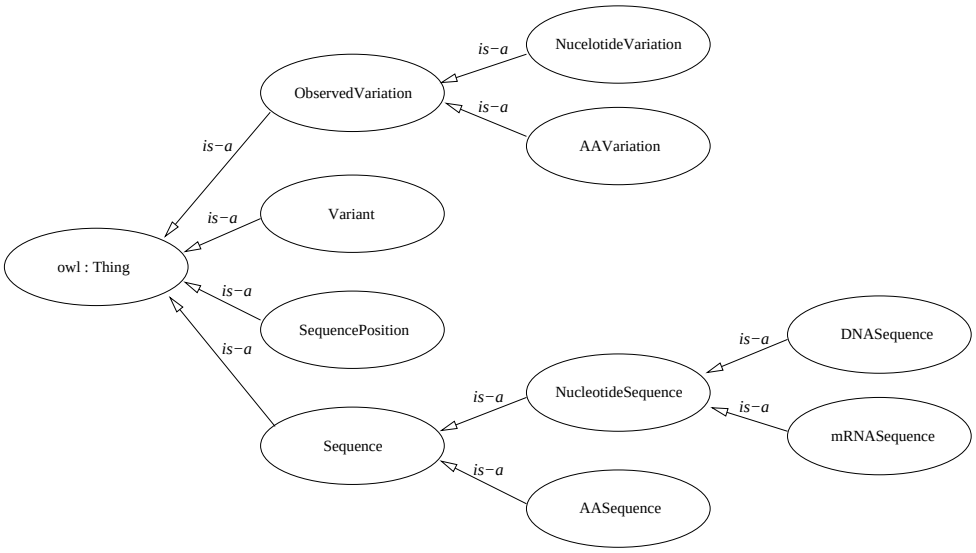

FIG. 3.5 – Représentation partielle de la hiérarchie de concepts de SNP-Ontology implémentée en OWL

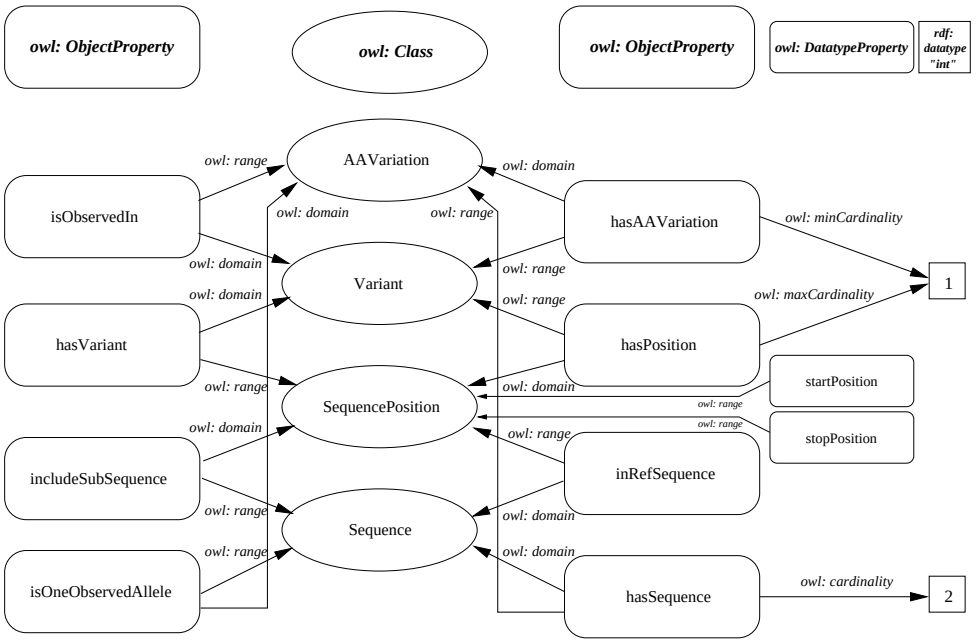

FIG. 3.6 – Représentation schématique de quelques concepts et rôles de SNP-Ontology implémentés en OWL. *N.B. :* en OWL, les concepts sont appelés des classes et les rôles sont soit des propriétés d'objets (*ObjectProperty*) soit des propriétés de type de données (*ObjectDataTypeProperty*). Les rôles présentent un domaine et un co-domaine (notés respectivement *owl :domain* et *owl :range*), et parfois une contrainte de cardinalité (*owl :minCardinality* par exemple).

mat particulier appelé OBO[53], il est donc nécessaire de la convertir en OWL. Cette conversion est réalisée à l'aide du plugin *BONG* de Protégé [WSGA03], puis validée manuellement.

1.2.5 Évaluation

SNP-Ontology et ses relations avec les ontologies externes sont consistantes.

Associée à un ensemble de wrappers développés spécialement et à l'application SNP-Converter décrite section 3.1.2, SNP-Ontology permet de répondre aux questions de compétence spécifiées.

SNP-Ontology permet de représenter les connaissances relatives aux variations génomiques qui peuvent être extraites de dbSNP, OMIM, PharmGKB, HapMap, et des bases de données locus spécifiques.

Ces résultats ne sont pas plus détaillés ici car l'utilisation de SNP-Ontology pour l'intégration de données relatives aux variations génomiques (section 3.1) illustre son évaluation.

1.2.6 Discussion

SNP-Ontology constitue une première représentation formelle des variations génomiques, mise à disposition via diverses bibliothèques d'ontologies, notamment le BioPortal. Sa disponibilité lui permet d'être réutilisée, discutée, et modifiée librement par les membres de la communauté des bio-ontologies.

Un autre atout de l'ontologie provient des choix faits lors de sa construction qui sont notamment : la prise en compte du contenu des principales bases de données de variations pour le choix des concepts et l'évaluation de sa capacité à être instanciée avec le contenu de ces bases. De ces choix résulte une relative facilité à établir des correspondances entre les données des bases d'une part et les concepts et rôles de l'ontologie d'autre part. Ce genre de correspondance est indispensable pour définir les mappings données-ontologie sur lesquels s'appuie le processus d'intégration de données à l'aide d'une ontologie décrit dans la section 2 de ce chapitre. Enfin, la définition de relations avec des concepts provenant d'ontologies externes permet de réutiliser de façon cohérente, dans le cadre de SNP-Ontology, l'ensemble des connaissances spécialisées élaborées par des consortiums d'experts comme le GO-consortium[54].

[53] Format OBO : `http://www.geneontology.org/GO.format.obo-1_2.shtml` (Dernière visite, le 27/07/2008)

[54] GO-consortium : `http://www.geneontology.org/GO.consortiumlist.shtml` (Dernière visite, le 27/07/2008)

1.3 Construction d'une ontologie pour la pharmacogénomique : SO-Pharm

1.3.1 Spécification

Domaine couvert par SO-Pharm SO-Pharm (pour *Suggested Ontology for Pharmacogenomics*) est une proposition de représentation formelle des connaissances pharmacogénomiques. SO-Pharm articule plusieurs ontologies des sous-domaines complémentaires de la pharmacogénomique, *i.e.* relatives aux génotype, phénotype, médicaments, et essais cliniques. Elle permet de représenter des relations pharmacogénomiques entre un médicament, une variation génomique, et un trait du phénotype. SO-Pharm permet de représenter également des patients, et plus généralement des panels impliqués dans des essais cliniques, et des populations. SO-Pharm permet de représenter les variables mesurées chez ces patients comme l'observation d'un phénotype ou le génotypage de variations génomiques. Elle inclut des connaissances relatives aux études de cas, à l'investigation clinique, et au test de nouvelles hypothèses en pharmacogénomique.

Objectifs de SO-Pharm SO-Pharm, comme SNP-Ontology, est conçue pour faciliter l'*intégration de données* et l'*extraction de connaissances en pharmacogénomique*. SO-Pharm est notamment développée pour pallier l'absence d'ontologie, elle regroupe dans une représentation cohérente les ontologies des sous-domaines de la pharmacogénomique.

Critères d'évaluation particuliers Des exemples de questions de compétence auxquelles SO-Pharm doit répondre sont :

- Un patient qui prend un traitement de codéine par voie orale avec une posologie de 50 mg trois fois par jours présente-t-il un risque de faire une réaction adverse ?
- Des troubles neurologiques peuvent-ils être une conséquence d'un traitement à la codéine ?
- Existe-t-il des variations génomiques du gène *CYP2D6* qui sont associées à l'absence d'effet analgésique en réponse à la codéine ?
- La réponse à un traitement de statines est il soumis à l'influence de facteurs génétiques ?

SO-Pharm doit permettre de représenter les connaissances pharmacogénomiques qui peuvent être extraites de OMIM et PharmGKB ainsi que des connaissances extraites de la littérature du domaine, par exemple les résultats rapportés dans [DGDM91, MTB+99, HVK+02, MMK+03].

Liste de sources de données et de connaissances relatives Dans le cas de SO-Pharm, les experts du domaine ont défini quatre listes de termes, relative chacune à la description d'un sous-domaine différent : génotype, phénotype, médicament, et essai clinique. La liste des sources de données et de connaissances, représentée Tableau 3.4, est établie pour enrichir les quatre listes de termes. Certaines sources ont été ajoutées au cours des différentes itérations du processus de construction de SO-Pharm. L'ajout d'une source peut amèner à l'ajout de nouveaux termes, de nouveaux concepts, et à l'articulation avec de nouvelles ontologies. Le Tableau 3.5 représente les sources de connaissances sélectionnées pour être articulées avec SO-Pharm.

1.3.2 Conceptualisation

Les trois Figures 3.7, 3.8, et 3.9 sont trois diagrammes de classes construits pour la conceptualisation de SO-Parm. Ils présentent respectivement la conceptualisation adoptée pour la notion d'*item clinique* (*i.e.* une donnée enregistrée relative à un patient), celle d'*essai clinique*, et celle de *protocole*. La Figure D.1 en Annexe D propose une vue plus générale du modèle conceptuel et permet notamment de situer les uns par rapport aux autres les trois diagrammes de classes présentés. La Figure 3.7 représente entre

Nom de la source	*Type de source*	*Domaine*	*URL*
dbSNP	Schéma XML, modèle de données	génotype	http ://www.ncbi.nlm.nih.gov/projects/SNP/
HapMap	Schéma XML	génotype	http ://www.hapmap.org/
HGVBase	DTD, modèle de données	génotype	http ://hgvbase.cgb.ki.se/
OMIM	Source de données	génotype, phénotype	http ://www.ncbi.nlm.nih.gov/omim/
OMG SNP	modèle de données	génotype	http ://www.omg.org/technology/documents/formal/snp.htm
MECV	Controlled vocabulary	génotype	http ://www.ebi.ac.uk/mutations/
SNP-Ontology	Ontologie OWL	génotype	
AA Ontology	Ontologie OWL	génotype	http ://www.co-ode.org/ontologies/amino-acid/
PharmGKB	Schéma XML, modèle de données	génotype, médicament, phénotype	http ://www.pharmgkb.org/
Pharmacogenetics Ontology	Vocabulaire contrôlé	genotype, phénotype	http ://www.pharmgkb.org/home/projects/project-po.jsp
Sequence Ontology	Vocabulaire contrôlé*	génotype	http ://song.sourceforge.net/
Gene Ontology	Vocabulaire contrôlé*	génotype	http ://www.geneontology.org/
PubChem	Source de données	médicament	http ://pubchem.ncbi.nlm.nih.gov/
RX-Norm	Vocabulaire contrôlé	médicament	http ://www.nlm.nih.gov/research/umls/rxnorm/index.html
ChEBI	Vocabulaire contrôlé*	médicament	http ://www.ebi.ac.uk/chebi/
CDISC	Schéma XML	phénotype	http ://www.cdisc.org/
ICD-10	Vocabulaire contrôlé	phénotype	http ://www.who.int/classifications/icd/
Disease Ontology	Vocabulaire contrôlé*	phénotype	http ://diseaseontology.sourceforge.net
Mammalian Phenotype	Vocabulaire contrôlé*	phénotype	http ://www.informatics.jax.org/searches/MP_form.shtml
PATO	Vocabulaire contrôlé*	phénotype	http ://obo.sourceforge.net/
Unit Ontology	Vocabulaire contrôlé*	phénotype	http ://obo.sourceforge.net/
Pathway Ontology	Vocabulaire contrôlé*	génotype, phénotype	http ://rgd.mcw.edu/tools/ontology
SNOMED-Clinical	Vocabulaire contrôlé	phénotype	http ://www.snomed.org/snomedct/glossary.html
Family Bond Ontology	Ontologie OWL	essai clinique	http ://www.loria.fr/∼coulet/ontology/familybond/version0.1/familybond.owl
Clinical Trial Ontology	Ontologie OWL	essai clinique	http ://www.bioontology.org/wiki/index.php/CTO :Main_Page
Ontology of Biomedical Investigations	Ontologie OWL	essai clinique	http ://obi.sourceforge.net/
OBO relationship types	Vocabulaire contrôlé*	méta-ontologie	http ://www.obofoundry.org/ro/
Basic Formal Ontology	Ontologie OWL	méta-ontologie	http ://www.ifomis.org/bfo

TAB. 3.4 – Liste des sources explorées pour enrichir la liste de termes relatifs aux sous-domaines de la pharmacogénomique. La troisième colonne précise le sous-domaine que la source concerne. Les vocabulaires contrôlés étoilés (*) sont des ontologies OBO.

autres les deux types principaux d'item cliniques : les items relatifs au génotype (Genotype item) et les items relatifs au phénotype (Phenotype item). Les premiers peuvent être des variants comme définis pour SNP-Ontology. Les seconds peuvent être composés à l'aide des concepts décrits pour l'ontologie PATO. La Figure 3.8 présente notamment qu'un item clinique (Clinical item) est mesuré durant un évenement (Clinical trial event) défini dans le cadre d'un essai clinique, est mesuré chez un individu (Individual),

Nom	*Description*	*Prefixe*	*Namespace*
SNP-Ontology	Variations génomiques	SNPO	~/ontology/snpontology/version1.5/snpontology_full.owl
Mutation Event Ont.	Classification des variations	MEO	~/ontology/meo/version1.0/meo.owl
AA Ontology	acides aminés	AAO	http ://www.co-ode.org/ontologies/amino-acid/2005/10/11/amino-acid.owl
Sequence Ontology	Séquences et variations	SO	http ://purl.org/obo/owl/SO
Pharmacogenetics Ont.	Méthodes de génotypage et de mesures	PGO	~/ontology/sopharm/version2.0/pharmacogeneticsontology.owl
Disease Ontology	Classification des maladies	DOID	~/ontology/sopharm/version2.0/diseaseontology.owl
Mammalian Phenotype	Critères relatifs au phénotype	MP	http ://purl.org/obo/owl/MP
PATO	Attributes et valeurs pour le phénotype	PATO	~/ontology/pato/version1.33/quality.owl
Unit Ontology	Unités de mesures	UO	~/ontology/unit/version1.9/unit.owl
ChEBI	Composé moléculaires	CHEBI	~/ontology/sopharm/version2.0/chebi.owl
Family Bond Ont.	Liens de parenté	FB	~/ontology/familybond/version0.1/familybond.owl
Clinical Trial Ontology	Protocole	CTO	http ://www.owl-ontologies.com/Ontology1178899652.owl
Ontology of Biomedical Investigation	Protocole	OBI	http ://obi.sourceforge.net/ontology/OBI.owl
Relationship Ontology	Types de relation	OBO_REL	http ://www.obofoundry.org/ro/ro.owl
Biomedical Function Ontology	Méta-ontologie	BFO	http ://www.ifomis.org/bfo/1.0

TAB. 3.5 – Les 15 ontologies articulées avec SO-Pharm. Le préfixe représenté par le symbole ~ correspond à l'URL http ://www.loria.fr/~coulet

et est mesuré selon une méthode (Measurement method) définie dans le cadre d'un protocole (Clinical trial protocole).La Figure 3.9 représente notamment qu'un protocole peut être composé d'un traitement médicamenteux (Drug treatment) composé d'un médicament (Drug) et d'une posologie (Posology) précise.

1.3.3 Formalisation

La formalisation des relations avec les concepts des ontologies sélectionnées est rapportée dans le Tableau 3.6.

1.3.4 Implémentation

SO-Pharm est disponible en OWL sur le Web à l'adresse suivante : `http://www.loria.fr/~coulet/sopharm2.0_description.php`.

La version 2.0 alpha contient 70 concepts dont 37 concepts définis et 56 rôles. En incluant les ontologies articulées avec SO-Pharm le nombre de concepts s'élève à 84786 et celui des rôles à 189. Ce nombre important de concepts est en grande partie dû au nombre élevé de concepts dérivés des vocabulaires spécialisés comme ChEBI ou Disease Ontology dont le nombre de termes atteint par exemple 15 192 pour la version 46 de ChEBI.

Concernant la conversion en OWL des ontologies articulées, elle dépend du format d'origine de chaque ontologie. Par exemple, sont disponibles en OWL et ne nécessitent donc aucune conversion SNP-Ontology, AA Ontology, CTO, OBI, BFO. Les ontologies disponibles dans le format OBO sont converties à l'aide du plugin *BONG* de Protégé [WSGA03], puis validées manuellement. Les ontologies

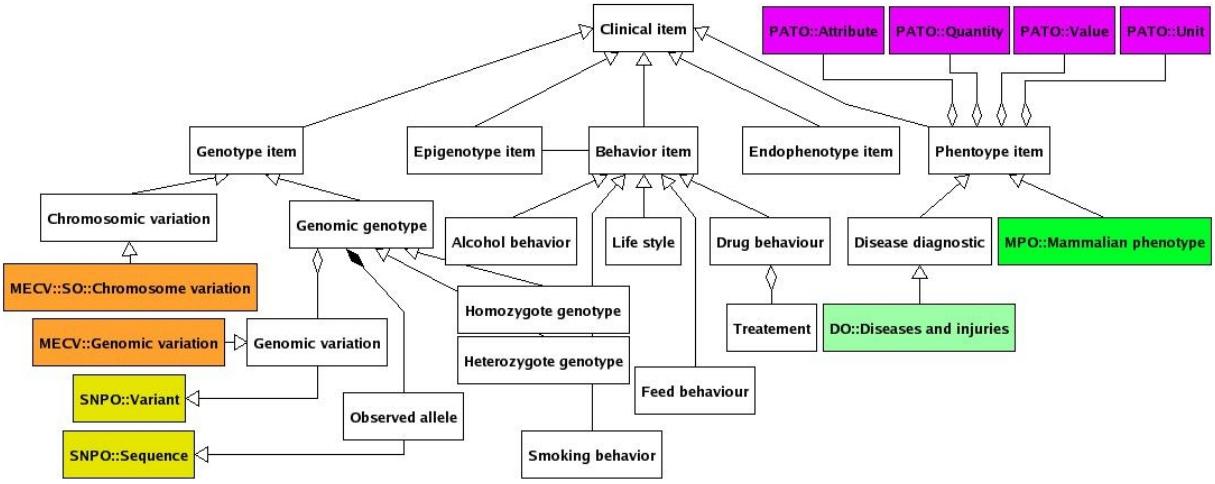

FIG. 3.7 – Diagramme de classes UML centré sur la conceptualisation des items cliniques

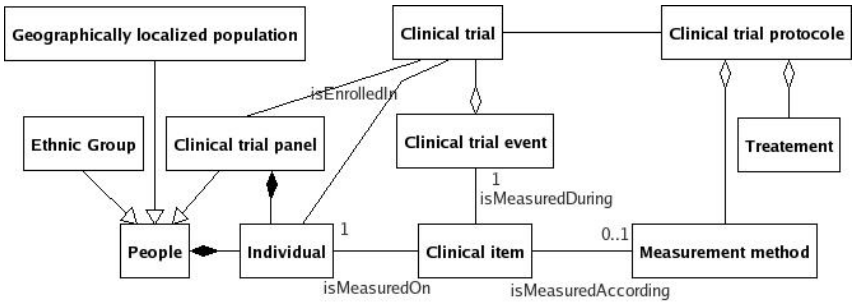

FIG. 3.8 – Diagramme de classes UML centré sur la conceptualisation d'essais cliniques

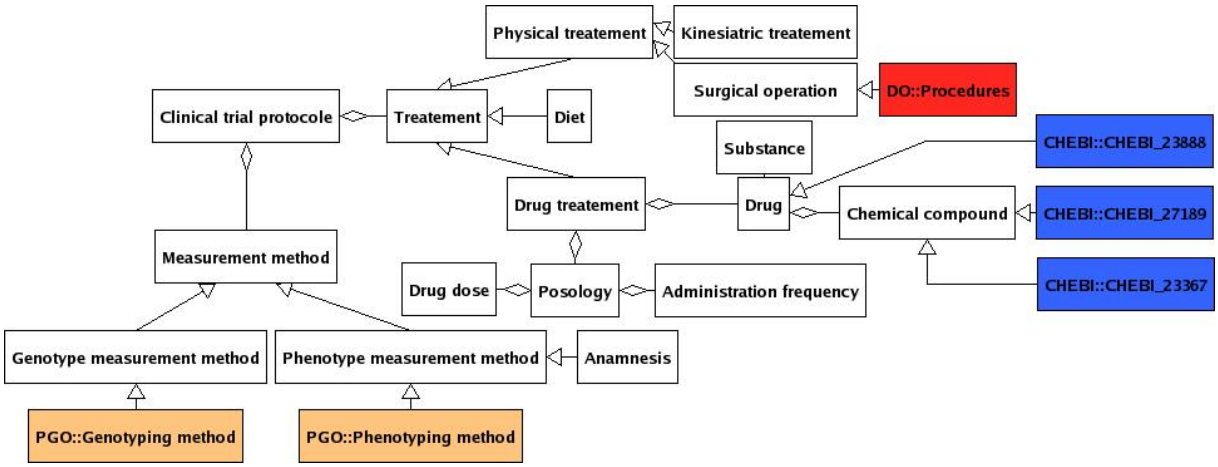

FIG. 3.9 – Diagramme de classes UML centré sur la conceptualisation d'un protocole d'essai clinique

SOPHARM :phenotype_item ⊒ MP :phenotype_ontology (MP :0000001)
SOPHARM :disease_diagnostic ⊒ DOID :disease_and_injuries (DOID :952)
SOPHARM :surgical_operation ⊒ DOID :procedures (DOID :1008)
SOPHARM :drug ⊒ CHEBI :drug (CHEBI :23888)
SOPHARM :chemical_compound ⊒ CHEBI :molecular_entities (CHEBI :23367)
SOPHARM :chemical_compound ⊒ CHEBI :unclassified (CHEBI :27189)
SOPHARM :chemical_compound ⊒ OBI :ChEBI_objects (OBI :263)
SOPHARM :chromosome_variation ⊒ SO :chromosome_variation (SO :0000240)
SOPHARM :genomic_variation ⊑ SNPO :variant
SOPHARM :genomic_variation ⊒ MEO :genomic_variation (MEO :001)
SOPHARM :observed_allele ≡ SNPO :sequence ⊓ ∀ isPartOf.SOPHARM :genomic_genotype
SOPHARM :population ⊒ SNPO :population
SOPHARM :genotype_measurement_method ⊒ PGO :genotyping_methods
SOPHARM :phenotype_measurement_method ⊒ PGO :phenotyping_methods
SOPHARM :phenotype_measurement_method ⊒ CTO :observations
SOPHARM :phenotype_item ⊑(∀ PATO :is_magnitude_of.PATO :quality ⊓ =1 PATO :is_magnitude _of) ⊔ (∀ PATO :is_measurement_of.PATO :quantitative ⊓ =1 PATO :is_measurement_of)
SOPHARM :phenotype_item ⊑ ∀ PATO :has_unit.UO :unit ⊓ =1 PATO :has_unit
SOPHARM :drug_dose ⊑ PATO :physical_quality ⊓ BFO :quality
SOPHARM :drug_dose ⊑ ∀ PATO :has_unit.UO :unit ⊓ =1 PATO :has_unit
SOPHARM :administration_frequency ⊑ PATO :frequency ⊓ BFO :quality
SOPHARM :administration_frequency ⊑ ∀ PATO :has_unit.UO :unit ⊓ =1 PATO :has_unit
SOPHARM :person ⊑ FB :family_member

Tab. 3.6 – Les principaux axiomes décrivant des relations entre les concepts propres à SO-Pharm (SOPHARM) et les concepts externes des ontologies articulées (voir Tableau 3.5). Les identifiants des concepts associés sont donnés entre parenthèses lorsqu'ils existent. La liste complète inclut également des axiomes qui formalisent des relations entre rôles.

disponibles sous d'autres formats sont converties manuellement. C'est le cas de l'ontologie *Pharmacogenetics Ontology*, disponible en HTML, ou de l'ontologie *Mutation Event Ontology* construite à partir du vocabulaire contrôlé *Mutation Event Controlled Vocabulary* et d'une partie de Sequence Ontology.

1.3.5 Évaluation

Le grand nombre de concepts articulés limite l'utilisation des mécanismes de raisonnement qui permettent la validation de la consistance et la classification des concepts. Les implémentations actuelles de ces mécanismes sont sensibles à la complexité de la LD utilisée (ici $\mathcal{SHOIN}(\mathcal{D})$) mais aussi au nombre de concepts de l'ontologie. Aussi pour valider la consistance et permettre la classification des concepts, sur une station de travail (CPU : Intel Pentium M 1.8GHz, RAM : 2 Go) nous avons utilisé les mécanismes de raisonnement sur l'ensemble des paires d'ontologies possibles (SO-Pharm – Disease Ontology puis SO-Pharm – ChEBI puis etc.).

Associée à un ensemble de wrappers développés spécialement, SO-Pharm permet de répondre aux questions de compétences spécifiées. L'utilisation de SO-Pharm dans le cadre d'extraction de connaissances en pharmacogénomique (voir section 2.4 du chapitre 4) permet notamment de mieux répondre à ces questions.

SO-Pharm permet de représenter les connaissances pharmacogénomiques qui peuvent être extraites

de OMIM et PharmGKB ainsi que des connaissances extraites de la littérature du domaine, par exemple les résultats rapportés dans [DGDM91, MTB+99, HVK+02, MMK+03]. SO-Pharm permet également de représenter de nouvelles hypothèses de connaissances pharmacogénomiques comme l'association entre une variation génomique, un traitement, et un ensemble de signes relevant d'un phénotype. L'utilisation de SO-Pharm dans l'objectif d'extraire des connaissances, décrite chapitre 4, a permi l'évaluation puis l'amélioration de l'ontologie.

1.3.6 Discussion

Au final, la construction manuelle de l'ontologie SO-Pharm propose une mise en correspondance cohérente de quinze ontologies sélectionnées. L'avantage est la maîtrise de la coexistence de concepts dont l'interprétation est équivalente, ou se recouvre de manière plus ou moins partielle, et surtout de manière plus ou moins ambiguë. La construction et la mise en correspondance manuelles demandent un effort important qui est justifié par la possibilité résultante de représenter des connaissances pharmacogénomiques en instanciant des relations existant entre plusieurs ontologies de sous-domaines et de raisonner sur ces connaissances de façon cohérente par les mécanismes de raisonnement classiques. De façon similaire à SNP-Ontology, SO-Pharm présente l'avantage de proposer à la communauté une première représentation formelle de son domaine avec l'objectif de faciliter sa réutilisation et son évolution. Pour aller dans ce sens, les dernières versions de SO-Pharm satisfont aux exigences de qualité proposées par l'OBO Foundry. Ces développements permettent à SO-Pharm de faire partie de l'OBO Foundry[55]. Des indications sur la façon dont SO-Pharm répond aux critères de cette forge particulière sont disponibles en ligne :

`http://www.loria.fr/~coulet/ontology/sopharm/version2.0/foundry_requirements.php`.

Il est intéressant de noter que certains de ces critères font débat et notamment le principe d'*orthogonalité* selon lequel le domaine recouvert par une nouvelle ontologie ne doit pas chevaucher celui des ontologies existantes dans la forge. Ce principe cherche à favoriser l'amélioration des ontologies existantes de façon communautaire plutôt qu'au développement d'ontologies concurrentes pour un même domaine. Ce point est discutable d'une part parce que la notion d'orthogonalité n'est pas définie de façon précise, et d'autre part parce qu'une ontologie est une représentation d'un domaine selon un point de vue particulier. Par conséquent, deux ontologies peuvent représenter selon deux points de vues différents un seul et même domaine. Pour cette raison les critères d'inclusion d'OBO-Foundry sont discutés au sein de la communauté et sont amenés à évoluer.

[55] `http://obofoundry.org/cgi-bin/detail.cgi?id=pharmacogenomics`

2 Intégration de données *guidée par* une ontologie

2.1 Description générale de l'approche proposée

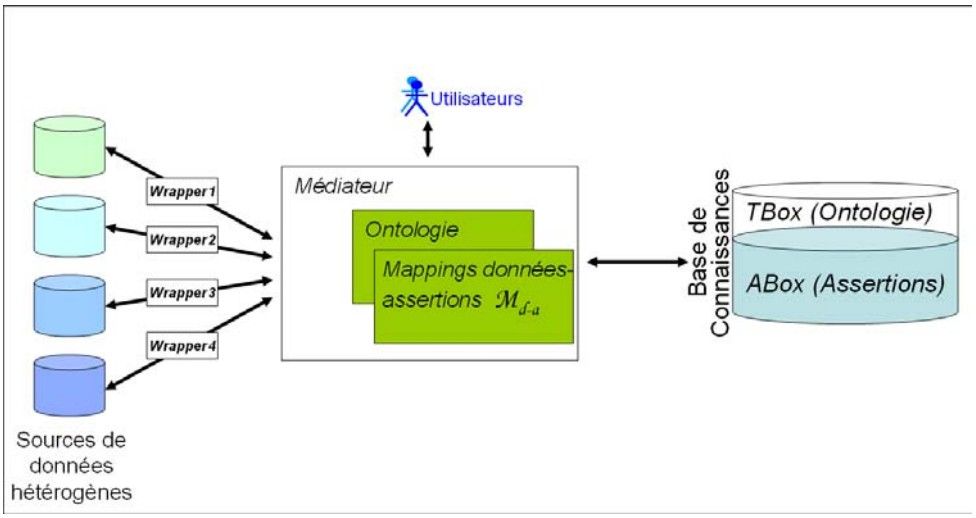

FIG. 3.10 – Architecture générale de notre système d'intégration de données. L'ontologie utilisée par le médiateur est la même que celle qui constitue la *TBox* de la Base de Connaissances.

Malgré l'existence d'architectures de référence ([CGL+98] par exemple), il n'existe pas d'architecture standard pour les systèmes d'intégration de données fondés sur une ontologie. L'architecture représentée Figure 3.10 que nous avons choisie peut être comparée à celle d'une approche médiateur comme décrit dans le chapitre 2 : les différentes sources sont mises en correspondance avec un vocabulaire global, dont la particularité ici est d'être une ontologie ; l'extraction des données est prise en charge par des *wrappers* et centralisée sous forme d'une réponse unique par le médiateur. Des mappings définis entre chaque source de données et l'ontologie permettent la traduction de requêtes pour l'interrogation des sources, puis en sens inverse, la traduction des réponses aux requêtes. C'est dans cette dernière phase que réside la distinction et l'apport majeur de notre approche. En effet, le médiateur élabore à l'aide des wrappers en réponse à une requête utilisateur : une liste d'assertions qui sert à instancier (ou peupler) la BC associée à l'ontologie.

Le déclanchement de l'intégration consiste en la soumission d'une requête par l'utilisateur. La requête initiale est décrite dans les termes de l'ontologies et le médiateur la traduit en requêtes sur les schémas locaux des sources de données. la traduction de la requête de l'utilisateur dans les termes des schémas locaux suit des approches déjà décrites [CGLV01, Len02], nous ne détaillons pas cette première phase. En revanche, les sections suivantes présentent plus amplement la façon dont sont définis les mappings entre les sources de données et l'ontologie, puis décrit l'interaction entre les wrappers et le médiateur.

Dans la suite de cette section, nous considérons chacune des sources comme une *base de données* possédants un schéma propre sur lequel il est possible d'exécuter des requêtes.

REMARQUE : Nous nous limitons ici à l'utilisation des bases de données, mais il pourrait être envisageable de développer des wrappers mettant en oeuvre des méthodes de Traitement Automatique de la Langue (TAL) pour peupler la BC.

2.2 Définition des mappings données–assertions

Pour chaque base de données considérée, la définition d'une requête dans les termes de son schéma et la transformation de la réponse à cette requête en une liste d'assertions s'appuient sur un mapping données-assertions [PLC+08]. Ces mappings sont définis au préalable, manuellement, et en considération des connaissances d'experts du domaine.

Définition 3.1 (Mapping données – assertions) *Soit un quadruplet* $(\mathcal{S}, \mathcal{M}_{d-a}, F, \mathcal{O})$ *où*

- $\mathcal{S}$ *est le schéma d'une base de données, i.e. un ensemble de relations n-aires de la forme* $R(A_1, A_2, ..., A_n)$ *et de domaine* $\prod_{i=1}^{n} D_i$ *tels que* A_i *est l'attribut d'indice i et de domaine* D_i.
- $\mathcal{O}$ *est une ontologie, i.e. les concepts d'un domaine et les rôles qui décrivent les relations entre ces concepts,*
- $\mathcal{M}_{d-a}$ *est un ensemble d'associations entre des données et des assertions dont chacune est de la forme*

$$\Phi \rightsquigarrow \Psi$$

où Φ *est une requête arbitraire sur la base de données de schéma* $\mathcal{S}$*, et* Ψ *est un ensemble d'assertions de concepts et d'assertions de rôles de l'ontologie* $\mathcal{O}$.

- *Enfin,* F *un ensemble de fonctions de la forme* $f_i(v)$ *applicables aux différentes valeurs résultant des requêtes* Φ *pour les transformer en noms d'individus dans* Ψ.

Les fonctions de F appliquées sur les valeurs des attributs sont définies de telle sorte que :

- deux valeurs d'attribut distinctes dans une ou plusieurs bases de données donnent lieu à deux noms d'individus distincts dans la BC,
- deux valeurs d'attributs, potentiellement distinctes, mais qui font référence à la même entité dans des bases de données différentes, donnent lieu à la création d'un seul et même nom d'individu,
- pour chaque mapping impliquant $f_i \in F$ il est possible de définir une fonction inverse notée f_i^{-1} qui permet à partir d'un identifiant d'individu de la BC de retrouver la valeur correspondante dans une base de données.

Les fonctions peuvent être définie soit manuellement, soit par des heuristiques. Comme l'illustre la suite de la thèse (chapitre 3, section 3.1 et chapitre 4 section 1), une fonction peut notamment être une composition d'autres fonctions ou prendre en compte les valeurs prises par d'autres attributs.

L'étape de peuplement de la BC associée à l'ontologie $\mathcal{O}$ revient à ajouter à la BC, pour l'ensemble des n-uplets réponses aux requêtes Φ, l'ensemble des assertions de concepts et des assertions de rôles Ψ du mapping $\mathcal{M}_{d-a}$ défini entre le schéma $\mathcal{S}$ et l'ontologie $\mathcal{O}$. Les individus impliqués dans les assertions du mapping qui n'existent pas encore dans la BC associée à $\mathcal{O}$ sont créés. De cette façon les fonctions appliquées aux valeurs d'attributs peuvent être utilisées pour nettoyer, transformer, homogénéiser le contenu des bases de données lors de l'instanciation.

Exemple Soit BD_1 et BD_2 deux bases de données dont les schémas $\mathcal{S}_1$ et $\mathcal{S}_2$ contiennent respectivement les deux relations suivantes R_1 et R_2 :

$$R_1\ (A_1, A_2, A_3)$$
$$R_2\ (A_1, B_2, B_3)$$

Dans cet exemple nous considérons que les attributs A_1 de R_1 et de R_2 sont identiques : ils ont le même nom, font référence au même concept, et leurs valeurs sont représentées en suivant la même syntaxe. Les attributs A_2 et B_2 font référence à un même concept, mais leurs valeurs sont représentées suivant des syntaxes différentes ce qui rend nécessaire l'utilisation de fonctions différentes (f_2 et f_4) pour qu'elles

soient transformées en identifiants d'individus qui suivent une syntaxe homogène. Les attributs A_3 et B_3 font référence à des concepts différents.

Deux exemples de définition de mapping possibles $\mathcal{M}_{d-a\ A}$ entre la base de données BD_1 et l'ontologie O, et $\mathcal{M}_{d-a\ B}$ entre BD_2 et la même ontologie O sont présentés ci-après sous la forme de l'association entre une requête SQL et une liste d'assertions en LD. Les notations sont inspirées de Poggi *et al.* [PLC+08]. En particulier on utilise la notation $f_1(A_1)$ pour représenter de façon générique le nom de chaque individu ce qui correspond à l'image de la fonction f_1 associée à chaque valeur prise par l'attribut A_1 dans la requête SQL :

$\mathcal{M}_{d-a\ 1}$:

```
SELECT  A1, A2, A3
FROM    R1
```

$\rightsquigarrow$ $ConceptUn(f_1(A_1))$, $ConceptDeux(f_2(A_2))$, $RoleUnVersDeux(f_1(A_1),\ f_2(A_2))$, $RoleUnVersDeux^{-}(f_2(A_2),\ f_1(A_1))$, $ConceptTrois(f_3(A_3))$, $RoleUnVersTrois(f_1(A_1),\ f_3(A_3))$, $RoleUnVersTrois^{-}(f_3(A_3),\ f_1(A_1))$

$\mathcal{M}_{d-a\ 2}$:

```
SELECT  A1, B2, B3
FROM    R2
WHERE   B3 ="aSpecificValue"
```

$\rightsquigarrow$ $ConceptUnSpecifique(f_1(A_1))$, $ConceptDeux(f_4(B_2))$, $RoleUnVersDeux(f_1(A_1),\ f_4(B_2))$, $RoleUnVersDeux^{-}(f_4(B_2),\ f_1(A_1))$, $ConceptQuatre(f_5(B_3))$, $RoleUnVersQuatre(f_1(A_1),\ f_5(B_3))$, $RoleUnVersQuatre^{-}(f_5(B_3),\ f_1(A_1))$

Suivant notre exemple, considérons les deux tuples suivants, réponses respectives aux requêtes de $\mathcal{M}_{d-a\ 1}$ et $\mathcal{M}_{d-a\ 2}$ sur BD_1 et BD_2 et les listes d'assertions qui leurs sont associées suivant les mappings :

$(a_1,a_2,a_3) \rightsquigarrow$ $ConceptUn$(a_1), $ConceptDeux$(a_2), $RoleUnVersDeux$(a_1, a_2), $RoleUnVersDeux^{-}$(a_2, a_1), $ConceptTrois$(a_3), $RoleUnVersTrois$(a_1, a_3), $RoleUnVersTrois^{-}$(a_3, a_1)

$(a_1,b_2,b_3) \rightsquigarrow$ $ConceptUnSpecifique$(a_1), $ConceptDeux$(a_2), $RoleUnVersDeux$(a_1, a_2), $RoleUnVersDeux^{-}$(a_2, a_1), $ConceptQuatre$(b_3), $RoleUnVersQuatre$(a_1, b_3), $RoleUnVersQuatre^{-}$(b_3, a_1)

Ainsi, les deux valeurs respectives a_2 et b_2 des deux attributs A_2 et B_2 sont transformés par les fonctions f_2 et f_4 en un même nom d'individu a_2, ce qui permet la création d'un seul individu identifié par a_2 et instance du concept *ConceptDeux* dans la BC :

$$ConceptDeux(a\_2)$$

Aussi, si

$$ConceptUnSpecifique \sqsubseteq ConceptUn,$$

le deuxième mapping apporte une nouvelle connaissance de par le fait que a_1 instancie non seulement *ConceptUn*, mais également *ConceptUnSpecifique*. Les assertions du rôle *RoleUnVersDeux* et de son inverse proposées par le deuxième mapping ($\mathcal{M}_{d-a\ 2}$) sont redondantes avec les assertions proposées par le premier mapping ($\mathcal{M}_{d-a\ 1}$). En conséquence, elles ne seront pas ajoutées à la BC. En revanche le deuxième mapping apporte une nouvelle connaissance en instanciant avec a_1 le rôle *RoleUnVersQuatre* et son inverse. Un exemple concret de mapping est donné dans ce chapitre en section 3.1.2.

Il est important de remarquer que la définition d'un mapping en collaboration avec l'expert nécessite l'existence, dans l'ontologie $\mathcal{O}$, des concepts et des rôles appropriés qui pourront être instanciés dans la BC. Si les concepts et le rôles adéquats n'existent pas, la définition du mapping constitue une motivation pour la mise à jour et l'amélioration de l'ontologie.

Comparé au triplet $(\mathcal{G}, \mathcal{S}, \mathcal{M})$ (associant un schéma global, les schémas des sources, et les mappings entre $\mathcal{G}$ et $\mathcal{S}$, voir section 3.1.2 du chapitre 2 et [Len02]) qui suffit à décrire les éléments de base d'un système d'intégration, notre approche inclut de façon supplémentaire un ensemble de fonctions qui garantit la transformation des valeurs en identifiants d'individus. Le fait que chaque ensemble de fonctions soit propre à une base de données et défini sans ambiguité permet que chaque fonction soit capable, inversement, de transformer un identifiant d'individu en une valeur de la base.

2.3 Description de l'interaction wrapper–médiateur

La première interaction entre médiateur et wrapper intervient lorsqu'un utilisateur émet une requête. Par exemple "*Quelles sont les variations génomiques et les médicaments associés à la maladie appelée Hypercholesterolémie Familiale*". Suivant le fonctionnement classique, le médiateur prend en charge la requête et l'adapte au schéma de chaque base de données. Les wrappers exécutent les requêtes adaptées aux différents schémas et récupèrent les données en réponse.

Ensuite, le médiateur permet, grâce aux mappings $\mathcal{M}_{d-a}$ (détaillés dans la définition 3.1), d'instancier la BC associée à l'ontologie en transformant de façon indépendante la réponse transmise par un wrapper en une liste d'assertions de concepts et d'assertions de rôles ajoutée à la BC. Les wrappers ne communiquent pas entre eux, mais le médiateur interagit avec la BC, et adapte ainsi l'instanciation au contenu de la BC qui se peuple progressivement. Si l'on reprend l'exemple de la requête relative à l'Hypercholestérolémie Familiale, lorsque le wrapper 2 extrait des données relatives à une variation génomique, il est possible que le médiateur ait déjà créé des individus relatifs à la même variation en conséquence des données transmises par le wrapper 1. Dans ce cas, le médiateur n'écrase pas les connaissances déjà disponibles dans la BC mais les complète si possible. Au final, le médiateur enchaîne une série d'instanciations cohérentes entre elles et avec l'ontologie pour intégrer les réponses successives des différentes bases de données.

2.4 Bilan

L'approche d'intégration de données proposée dans cette section s'inspire amplement *(1)* des architectures classiques des systèmes d'intégration de données [Hal01, CG05] et *(2)* de résultats théoriques décrit récemment sur la formalisation des mappings données–ontologies [PLC+08]. La principale originalité proposée ici est d'utiliser et d'adapter ces résultats théoriques au cadre d'une architecture opéra-

tionnelle qui peut ainsi articuler ainsi à la fois base de données et Base de Connaissances.

L'approche proposée a comme principal inconvénient qu'elle nécessite, pour chaque source, de définir un mapping données–assertions adapté, et de développer le wrapper associé. En contre-partie, cette méthode bénéficie des avantages de l'approche médiateur en terme d'indépendance vis à vis des sources : de nouvelles sources peuvent être intégrées sans que l'ontologie ne soit transformée. Cependant, si une source contient des données encore non considérées qu'il se révèle intéressant d'intégrer, l'ontologie peut nécessiter d'être enrichie par l'addition de concepts, rôles, axiomes de telle sorte que les nouvelles données puissent correspondre à des assertions de la BC.

Une autre limite provient des technologies actuelles de gestion de BC. Les opérations de raisonnement et notamment d'interrogation sur une BC sont problématiques lorsque la *TBox* ou la *ABox* deviennent trop volumineuses. Cette limite est accentuée lorsque le langage de représentation des connaissances est d'une expressivité plus importante et les mécanismes de raisonnement plus complexes. Notre approche évite le peuplement d'une BC trop volumineuse comme cela pourrait être le cas par une approche entrepôt. Ainsi une requête très spécifique, dont la réponse contient un nombre de tuples restreint, entraîne la constitution d'une BC tout aussi spécifique et peu volumineuse. Une requête plus générale donnera une réponse dotée de plus de tuples et constituera une BC également plus générale et plus volumineuse. En revanche, notre approche permet d'intégrer successivement les réponses de différentes requêtes dans la même BC dont le contenu s'élargira au fur et à mesure. De ce point de vue, notre approche présente certains des avantages des approches d'intégration type entrepôt puisque la BC peuplée par une ou plusieurs requêtes bénéficie d'une part de l'intégration de données, et d'autre part de la sémantique associée aux données.

Le fait de disposer des données intégrées sous forme d'assertions dans une BC nous intéresse particulièrement puisque cela permet, tout d'abord de représenter des relations qui ne peuvent pas l'être dans le cadre d'une base de données relationnelle classique, comme par exemple représenter le fait que deux représentations distinctes (par exemple de deux variations génomiques) font référence à une seule et même entité. Cela permet également, à l'aide des mécanismes de raisonnement, de valider la consistance du modèle, de classifier les individus de l'ontologie. Enfin, comme nous l'exposons dans le chapitre 4, la sémantique associée à la BC peut être utilisée pour guider l'extraction de connaissances implicites ou nouvelles et potentiellement utiles, par exemple en utilisant des méthodes de fouille de données sur les assertions de la BC.

Les deux sections suivantes (3.1 et 3.2) illustrent l'utilisation, pour l'intégration de données, des deux ontologies dont la construction est décrite en section 1.

3 Expérimentation

Cette section présente les résultats d'implémentation et de mise en œuvre de l'approche proposée section 2 pour l'intégration de données *guidée par* une ontologie. Les résultats rapportés ont été obtenus dans le cadre d'expérimentation sur des données relatives aux variations génomiques tout d'abord, puis à la pharmacogénomique.

3.1 Intégration de données relatives aux variations génomiques : SNP-Converter

La section 2.3 du chapitre 1 et notamment sa Figure 1.2 illustre les nombreuses façons de désigner de façon unique une variation génomique dans les bases de données publiques et privées. Il est important de noter que certaines notations non-conventionnelles (regroupées sous la section c dans la Figure 1.2) sont ambiguës : la première description ne mentionne pas le nucléotide de référence, la troisième et la quatrième font référence à deux versions différentes de la même protéine sans préciser de quelle version il s'agit.

L'évaluation précise du recouvrement entre les bases de données de variations génomiques est cruciale dans le cadre du développement de diagnostics génétiques et de l'exploration du *variome* (*i.e.* l'ensemble des variations du génome humain) [dDP03, RKC06, Spe08]. Cette tâche est rendue particulièrement délicate à cause du nombre important de représentations différentes et pourtant équivalentes. Aussi un système capable d'établir cette équivalence est nécessaire pour des investigations impliquant l' analyse de variations génomiques, et de cette façon est nécessaire comme base à une exploration avancée de la pharmacogénomique qui prend en considération les nombreuses données recueillies dans le domaine [AK02].

3.1.1 Les solutions d'intégration existantes

Une première solution au problème de la représentation hétérogène des variations consiste en la construction d'une base de données unique qui permette un accès à l'ensemble des variants contenus initialement dans différentes sources. C'est l'objectif de la base de données dbSNP du NCBI qui est la plus grande source de variations disponible sur le Web (voir la section 2.2 du chapitre 1). En plus de contenir les variations qui lui sont directement soumises, dbSNP intègre des données provenant d'autres grandes bases de données de variations génomiques comme la base NCI CGAP-GAI, HGVBase, HapMap, Perlgen. Un avantage stratégique de dbSNP est de faire partie des bases de données du NCBI (entre autres GenBank, PubMed, Gene, Human Genome Project Data) et à ce titre d'être interrogeable par le système fédéré *Entrez* [Bax06]. Un inconvénient de dbSNP est de ne pas permettre la coexistence de données publiques et de données privées relatives à des variations que les biologistes ne souhaitent pas diffuser (par exemple une nouvelle variation ou une nouvelle annotation).

TAMAL (*Time and Money are Limiting*) [HSS06] et *LS-SNP* (*Large-Scale annotation of coding non-synonymous SNPs*) [KDK+05] sont des systèmes d'intégration de données alternatifs, principalement basés sur le contenu de dbSNP, mais dont l'avantage est de proposer des annotations supplémentaires et des facilités de sélection de SNP d'intérêt pour la conception d'études cliniques. Ces SNP d'intérêt peuvent être les SNP susceptibles d'être associés à une maladie et donc intéressants à génotyper chez les patients enrôlés. Ces deux systèmes partagent l'inconvénient de dbSNP qui est de ne pas permettre l'intégration de données tierces.

3.1.2 SNP-Converter : un système de conversion et d'intégration de variations génomiques

SNP-Converter est un outil original développé pour l'intégration de données relatives aux variations génomique en suivant l'approche décrite section 2 (voir Figure 3.11). SNP-Converter utilise l'ontologie

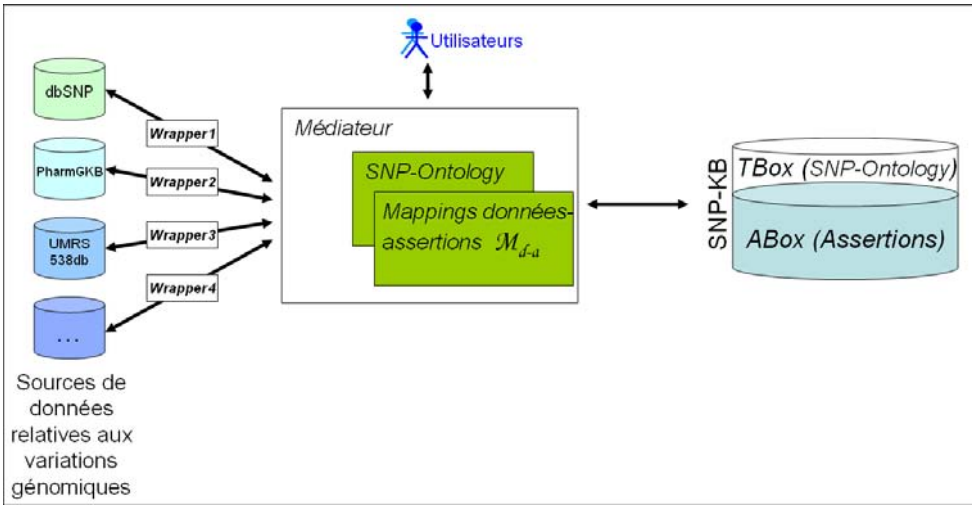

FIG. 3.11 – Architecture de SNP-Converter suivant celle proposée Figure 3.10

SNP-Ontology pour représenter, par un ensemble d'assertions de concepts et de rôles, n'importe quel variant quelle que soit sa description initiale. Grâce à cette capacité, des données contenues dans des sources hétérogènes peuvent être mises correspondance avec les concepts et rôles de SNP-Ontology par l'intermédiaire de mappings données–assertions. Suivant ces mappings, SNP-Converter permet le peuplement d'une BC associée à SNP-Ontology et appelée SNP-KB.

Tels qu'ils sont décrits dans la définition 3.1, les mappings données-assertions sont associés à un ensemble de fonction F qui assure la transformation des valeurs des bases de données en noms d'individus qui viennent peupler la BC. De part l'hétérogénéité des données relatives aux variations génomiques, cet ensemble de fonction est particulièrement important dans SNP-Converter puisqu'elles sont utilisées pour réaliser la *conversion* de la description d'une variation génomique en une autre. Ensuite, l'*intégration* proprement dite est effectuée par SNP-Converter qui est capable d'estimer l'équivalence entre deux descriptions converties en un format *pivot*, *i.e.* un *jeu de quatre attributs* (décrit ci-après) qui identifie de façon unique une variation génomique.

Réaliser la conversion de la description d'une variation génomique en une autre ou établir l'équivalence entre deux descriptions sont des opérations qui font intervenir des connaissances explicites du domaine : des connaissances relatives à la structure du gène, la définition d'un transcrit, ou encore au code génétique. L'une des raisons qui a motivé la construction de SNP-Ontology était justement de fournir une représentation de ces connaissances sur laquelle s'appuyer afin de permettre la conversion, la comparaison et au final l'intégration de ce type de données.

Un variant est une *variation observée* localisée sur une *position* précise le long d'une *séquence*. La variation observée peut être une variation de nucléotides ou d'acides aminés selon que la séquence qui sert de référence à sa localisation est un acide nucléique (*i.e.* ADN ou ARN) ou une protéine. Cette définition reflète à la fois le standard proposé par la nomenclature HGVS et la conceptualisation de SNP-Ontology. Elle implique qu'une variation soit décrite au minimum par un *jeu de quatre attributs* :

(i) l'identifiant d'une séquence de référence (*i.e.* son numéro d'accession dans une base de données publique) ;
(ii) le type de la séquence en question : génomique, codante/ADN_c, ARN_m, ou protéine respectivement abrégé par les lettres g., c., r., p. suivant le standard de l'HGVS ;
(iii) la position du variant sur la séquence de référence ;
(iv) la variation observée (G/T, G/-, -/T, GT/AG, g/u, Gly/Val par exemple).

La conjonction de ces quatre attributs permet une description univoque du variant.

Comme mentionné dans la section 2.3, un même variant peut être décrit par différentes compositions de ce jeu de quatre attributs selon la séquence de référence choisie. Le principe général du SNP-Converter est de prendre en entrée un jeu d'attributs, et de le convertir en un jeu d'attributs alternatif qui représente le même variant.

SNP-Converter pour la conversion de format

Le processus mis en œuvre par SNP-Converter lors de la conversion de la description d'une variation peut être décomposé en quatre étapes détaillées dans la suite de cette section et illustrées par les Figures 3.12 et 3.13.

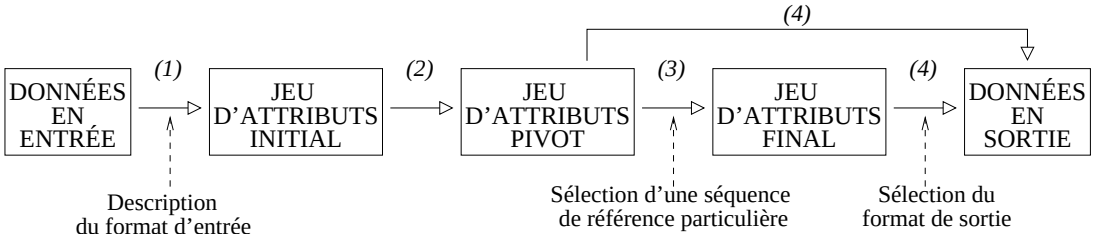

FIG. 3.12 – Les différentes étapes du processus de conversion de la description d'une variation génomique pris en charge par SNP-Converter

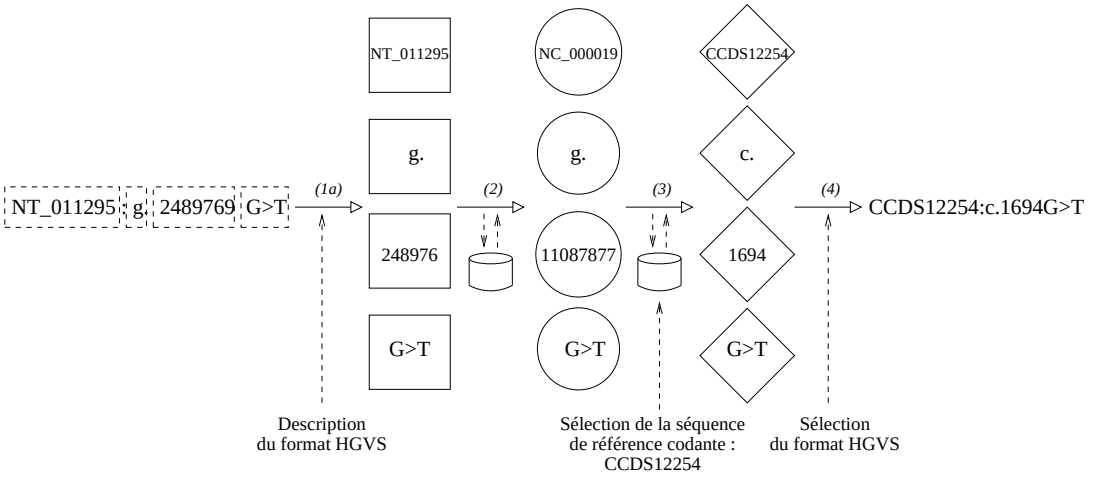

FIG. 3.13 – Exemple de conversion de la description d'une variation génomique réalisée par SNP-Converter

(1) L'étape de préparation.

Certaines descriptions ne décrivant pas explicitement les quatre attributs, il est nécessaire d'inclure dans l'application une étape de préparation. Cette étape consiste en l'extraction des quatre attributs initiaux et, en conséquence, est spécifique à chaque format de source de données. L'étape de préparation est différente selon que la description du variant est *explicite* (comme la syntaxe HGVS ou la syntaxe similaire à celle du genome-browser) ou *implicite* (un identifiant de base de données). Quand la description est explicite *(1a)*, les quatre attributs peuvent être directement obtenus en parcourant la description et en en extrayant chacun des attributs. Quand la description

est implicite *(1b)*, les attributs initiaux sont obtenus par une requête sur la base de données concernée. Par exemple, si la description de départ est un identifiant dbSNP, il est utilisé durant l'étape de préparation pour interroger dbSNP et extraire le jeu d'attributs initial. L'étape de préparation permet également de compléter une description *ambiguë (1c)*, soit en complétant automatiquement les données manquantes d'une base de données lorsque c'est possible, soit en complétant manuellement la description.

L'implémentation actuelle de cette étape de préparation permet l'extraction des quatre attributs à partir des entrées de dbSNP, HGVBase, HapMap, PharmGKB et de fichiers plats de deux bases de données privées qui suivent des représentations non-conventionnelles correspondant aux deux premiers exemples de la section c de la Figure 1.2.

(2) La conversion du jeu d'attributs initial en un *jeu pivot*.

Le jeu d'attributs pivot consiste en une version particulière des quatre attributs pour laquelle l'identifiant de la séquence de référence est celui de la séquence complète du chromosome (*i.e.* un numéro d'accession RefSeq de la forme NC_000019.8) qui contient la variation. En conséquence, le type de séquence dans le jeu pivot est génomique. Les deux attributs restant doivent quant à eux être calculés. La position relative de la séquence de référence initiale sur la séquence complète du chromosome est recherchée dans la base de données adaptée. Par exemple, la position relative d'un gène peut être trouvée à partir du symbole du gène dans l'entrée RefSeq du chromosome complet (dans la section "FEATURES/gene"). La position génomique des exons peut également y être retrouvée dans la section "FEATURES/mRNA". Si la position du variant est donnée par rapport au début de la séquence traduite, *i.e.* du codon *start* ATG, les coordonnées des morceaux de séquences codantes peuvent être trouvées dans la base de données CCDS[56] du NCBI. La position exacte de la variation sur la séquence complète du chromosome peut être calculée à partir de ces données et de la position de la variation sur la séquence de référence initiale. Enfin l'attribut correspondant à la variation observée doit être converti en une variation de séquence génomique. Si la variation observée est initialement décrite sur une séquence d'ADN, elle reste identique sauf dans le cas exceptionnel où elle est observée sur le brin anti-sens auquel cas elle est convertie. Autrement, si la variation observée l'est sur une séquence d'ARN, les uraciles (U) doivent être convertis en thymines (T). Une variation observée au niveau d'une protéine est convertie suivant le code génétique. En raison de la dégénérescence du code génétique plusieurs codons codent pour le même acide aminé, ainsi la conversion acide aminé → nucléotide peut générer plusieurs propositions de variations. SNP-Converter génère toutes les possibilités.

(3) La conversion optionnelle en un jeu d'attributs final.

Cette conversion est optionnelle puisque dans le cas où la description désirée correspond au jeu pivot, elle est inutile. Cela est notamment le cas dans le processus d'intégration de données que nous détaillons par la suite qui se base sur le jeu pivot. Si ce n'est pas le cas, l'utilisateur doit sélectionner une séquence de référence d'ADN, ARN_c, ARN_m ou protéique sur laquelle doit être positionnée la variation. Le processus de conversion suit alors exactement le même raisonnement que pour la conversion précédente afin de déterminer la nouvelle position relative et la variation observée en fonction de la séquence de référence choisie.

(4) Le formatage des données de sortie.

Dans le cas de l'intégration de données, illustrée dans la section suivante, cette dernière étape consiste en la transformation du jeu d'attributs en un ensemble d'assertions en LD qui viendront instancier une BC. Cependant, SNP-Converter peut être utilisé comme simple convertiseur de format, indépendemment de tout système d'intégration. Dans ce cas, les données de sorties peuvent

[56] `http://www.ncbi.nlm.nih.gov/CCDS/`

être formatées selon l'usage qu'il est prévu d'en faire. Un premier choix peut être l'édition simple du jeu d'attributs final suivant la syntaxe HGVS. Un second choix est la création d'un fichier contenant la description de la variation dans le format spécifique de soumission à une base de données, comme par exemple le format XML de soumission à dbSNP.

SNP-Converter a donné lieu au développement d'un prototype en java dont plusieurs copies d'écran sont présentées en Annexe E. A l'aide de cette implémentation, SNP-Converter a été expérimenté sur les variations du gène *LDLR* contenues dans dbSNP (au format XML) et de variations du même gène, décrites de façon non-conventionnelle, dans des sources privées sous forme de fichiers textes. L'objectif était alors de mesurer le taux de recouvrement entre les trois bases de données et plus spécifiquement d'identifier les variations des bases de données privées qui ne sont pas enregistrées dans dbSNP, afin d'envisager leur soumission.

Pour réaliser cette expérimentation, SNP-Converter a d'abord été utilisé pour convertir les variations contenues dans les trois sources en leur description par le jeu pivot, pour ensuite comparer les résultats et évaluer leur équivalence potentielle. La fonction du SNP-Converter permettant d'instancier une BC a été utilisée pour intégrer les différentes descriptions de variations et leurs équivalences. L'instanciation de la BC et les résultats obtenus sont présentés dans la section suivante.

SNP-Converter pour l'intégration de données

L'utilisation du SNP-Converter pour l'intégration peut être considérée comme un mapping indirect entre le schéma des sources de données initiales et l'ontologie. Dans ce sens, le mapping indirect s'appuie alors sur un ensemble de fonctions de convertion des descriptions hétérogènes. Dans l'optique d'intégrer un maximum de données relatives aux variations, nous utilisons SNP-Converter de telle sorte que lorsqu'il instancie un nouveau variant dans la BC, il lui associe non seulement les attributs du jeu initial, mais également les attributs pivots calculés par SNP-Converter. Dans la même optique de l'intégration d'un maximum de données, il est également intéressant pour chaque variation d'intégrer dans la SNP-KB d'une part le jeu des quatre attributs et d'autre part des attributs supplémentaires associés à la variation en question (par exemple l'organisme étudié, ou sa fréquence d'observation dans une population). Dans ce cas les attributs supplémentaires sont extraits au même titre que ceux du jeu d'attributs considéré, mais ne sont soumis à aucune conversion. En revanche, pour qu'ils puissent donner lieu à l'instanciation de la BC il faut qu'ils soient inclus dans la description du mapping données-assertions (voir section 2.2).

Si l'on considère les deux bases de données *PharmGKB* et *dbSNP* dont les schémas contiennent respectivement les deux relations suivantes $R_{PharmGKB}$ et R_{dbSNP} :

$R_{PharmGKB}$ (Submission_Id, GP_Position, assembly, Strand, Variant, Feature, Nb_Of_Chr, Frequency, gene_symbole)

R_{dbSNP} (dbSNP_Id, organism, genome_build, alleles, contig_accession, contig_position, function, gene_symbole)

Deux exemples de mapping $\mathcal{M}_{d-a}$ (voir définition 3.1) possibles entre ces bases de données et l'ontologie *SNP-Ontology* $\mathcal{M}_{d-a\ 1}$ et $\mathcal{M}_{d-a\ 2}$ sont définis ici par l'association entre une requête SQL et des assertions en LD :

$\mathcal{M}_{d-a\ 1}$:

SELECT Submission_Id, GP_Position, Variant
FROM $R_{PharmGKB}$

$\rightsquigarrow$

$Variant(f_1(\texttt{Submission\_Id}))$,
$Position(f_2(\texttt{GP\_Position}))$,
$hasPosition(f_1(\texttt{Submission\_Id}),\ f_2(\texttt{GP\_Position}))$,
$hasPosition^{-}(f_2(\texttt{GP\_Position}),\ f_1(\texttt{Submission\_Id}))$,
$Variation(f_3(\texttt{Variant}))$,
$hasVariation(f_1(\texttt{Submission\_Id}),\ f_3(\texttt{Variant}))$,
$hasVariation^{-}(f_3(\texttt{Variant}),\ f_1(\texttt{Submission\_Id}))$

$\mathcal{M}_{d-a\ 2}$:

SELECT dbSNP_Id, contig_position, alleles
FROM R_{dbSNP}
WHERE function ="non-synonymous"

$\rightsquigarrow$

$NonSynonymousVariant(f_4(\texttt{dbSNP\_Id}))$,
$Position(f_5(\texttt{contig\_position}))$,
$hasPosition(f_4(\texttt{dbSNP\_Id}),\ f_5(\texttt{contig\_position}))$,
$hasPosition^{-}(f_5(\texttt{contig\_position}),\ f_5(\texttt{dbSNP\_Id}))$,
$Variation(f_6(\texttt{alleles}))$,
$hasVariation(f_4(\texttt{dbSNP\_Id}),\ f_6(\texttt{alleles}))$,
$hasVariation^{-}(f_6(\texttt{alleles}),\ f_4(\texttt{dbSNP\_Id}))$

Chaque variant, réponse à l'une des deux requêtes précédentes, est converti par SNP-Converter (SC) en quatre valeurs correspondant au jeu d'attributs pivot. Il est alors possible d'appliquer à ce jeu d'attributs particulier le mapping appelé $\mathcal{M}_{d-a\ SC}$ dont un exemple est présenté ci-après. Les fonctions de la forme sc_i représentent alors les opérations de conversion réalisées sur les valeurs des attributs du jeu initial. Le résultat de ces fonctions constitue le jeu d'attributs pivot. Respectivement sc_1 extrait l'identifiant de la séquence de référence, sc_2 la position sur cette séquence, sc_3 le type de la séquence de référence, et sc_4 la variation observée. L'exemple proposé de $\mathcal{M}_{d-a\ SC}$ présente la particularité que référence, position, et type de séquence soient extraits à partir du même attribut GP_Position. Les fonctions f_i sont les fonctions classiquement définies dans le cadre des mappings. La fonction f_7 présente la particularité de prendre 4 attributs en paramètre car elle construit un identifiant unique de variant sur la base des valeurs des quatre attributs du jeu pivot. Dans un souci de clarté nous remplacerons dans le mapping la notation :

$$f_7(sc_1(\texttt{GP\_Position}), sc_2(\texttt{GP\_Position}), sc_3(\texttt{GP\_Position}), sc_4(\texttt{Variant})) = f_7(\texttt{jeu\_pivot}).$$

SELECT Submission_Id, GP_Position, Variant
FROM $R_{PharmGKB}$

SC $\rightarrow$ $\mathcal{M}_{d-a\ SC}$:
$sc_1(\texttt{GP\_Position})$
$sc_2(\texttt{GP\_Position})$
$sc_3(\texttt{GP\_Position})$
$sc_4(\texttt{Variant})$

$\rightsquigarrow$

$Variant(f_7(\texttt{jeu\_pivot}))$,
$Sequence(f_8(sc_1(\texttt{GP\_Position})))$,
$isLocatedOn(f_7(\texttt{jeu\_pivot}),\ f_8(sc_1(\texttt{GP\_Position})))$,
$isLocatedOn^{-}(f_8(sc_1(\texttt{GP\_Position})),\ f_7(\texttt{jeu\_pivot}))$,
$Position(f_9(sc_2(\texttt{GP\_Position})))$,
$hasPosition(f_7(\texttt{jeu\_pivot}),\ f_9(sc_2(\texttt{GP\_Position})))$,
$hasPosition^{-}(f_9(sc_2(\texttt{GP\_Position})),\ f_7(\texttt{jeu\_pivot}))$,
$Variation(f_{10}(sc_4(\texttt{Variant})))$,
$hasVariation(f_7(\texttt{jeu\_pivot})))$,\ $f_{10}(sc_4(\texttt{Variant}))$,
$hasVariation^{-}(f_{10}(sc_4(\texttt{Variant})),\ f_7(\texttt{jeu\_pivot})))$

Dans le cas du second mapping $\mathcal{M}_{d-a\ 2}$, les attributs de R_{dbSNP} pris en paramètre par les fonctions sc_i sont différents, mais le mapping vers les assertions est identique.

L'étape d'instanciation de la BC SNP-KB revient à

– *(1)* ajouter, pour l'ensemble des n-uplets réponses aux requêtes, l'ensemble des assertions de concepts et des assertions de rôles du mapping $\mathcal{M}_{d-a}$ défini entre SNP-Ontology et le schéma $\mathcal{S}$ de la bases de données considérée ;

- *(2)* à partir des n-uplets réponses aux requêtes, extraire et convertir les valeurs en celles correspondant au jeu d'attributs pivot ;
- *(3)* ajouter pour l'ensemble des quadruplets résultant, l'ensemble des assertions du mapping $\mathcal{M}_{d-a\ SC}$;
- *(4)* enfin, définir dans la BC, l'équivalence entre le variant décrit par ses attributs initiaux et le variant décrit par les attributs pivot.

Le fait que SNP-Converter instancie dans la BC également le jeu pivot, permet de tester l'équivalence de deux variants dont les descriptions initiales étaient différentes mais dont la description pivot est identique. Le test d'équivalence peut être considéré comme une extension procédurale des mécanismes de raisonnement classiques. Le résultat de ce test aboutit à l'enrichissement de la BC.

Pour terminer notre exemple, considérons les deux tuples suivants, réponses respectives aux requêtes de $\mathcal{M}_{d-a\ 1}$ et $\mathcal{M}_{d-a\ 2}$ sur *dbSNP* et *PharmGKB* et les assertions associées :

(135411387,Chr6 :18247207,A/G)	$\rightsquigarrow$	*Variant*(135411387_01), *Position*(Chr6_18247207), *hasPosition*(135411387, Chr6_18247207), *hasPosition*$^-$(Chr6_18247207, 135411387), *Variation*(A_G), *hasVariation*(135411387_01, A_G), *hasVariation*$^-$(A_G, 135411387_01)
(rs1800460,8997479,G>A)	$\rightsquigarrow$	*NonSynonymousVariant*(rs1800460_01), *Position*(8997479), *hasPosition*(rs1800460_01, 8997479), *hasPosition*$^-$(8997479, rs1800460_01), *Variation*(A_G), *hasVariation*(rs1800460_01, A_G), *hasVariation*$^-$(A_G, rs1800460_01)

Ainsi, les deux attributs 'A/G' et 'G>A' sont transformés par les fonctions f_3 et f_6 en un même nom d'individu 'A_G', et permettent ainsi la création d'un seul individu identifié par 'A_G' qui est instance du concept *Variation* dans la BC.

		$\mathcal{M}_{d-a\ SC}$		*Variant*(ch6_18247207_c_A_G), *Sequence*(NC_000006),
(135411387, Chr6 :18247207, A/G)	SC →	sc_1(Chr6 :18247207) sc_2(Chr6 :18247207) sc_3(Chr6 :18247207) sc_4(A/G)	⇝	*isLocatedOn*(NC_000006, ch6_18247207_c_A_G), *isLocatedOn*$^-$(ch6_18247207_c_A_G, NC_000006), *Position*(18247207), *hasPosition*(ch6_18247207_c_A_G, 18247207), *hasPosition*$^-$(18247207, ch6_18247207_c_A_G), *Variation*(A_G), *hasVariation*(ch6_18247207_c_A_G, A_G), *hasVariation*$^-$(A_G, ch6_18247207_c_A_G)
		$\mathcal{M}_{d-a\ SC}$		*Variant*(ch6_18247207_c_A_G), *Sequence*(NC_000006),
(rs1800460, 8997479, G>A)	SC →	sc_1(NT_007592) sc_2(8997479) sc_3(NT_007592) sc_4(G>A)	⇝	*isLocatedOn*(NC_000006, ch6_18247207_c_A_G), *isLocatedOn*$^-$(ch6_18247207_c_A_G, NC_000006), *Position*(18247207), *hasPosition*(ch6_18247207_c_A_G, 18247207), *hasPosition*$^-$(18247207, ch6_18247207_c_A_G), *Variation*(A_G), *hasVariation*(ch6_18247207_c_A_G, A_G), *hasVariation*$^-$(A_G, ch6_18247207_c_A_G)

Les deux variants exemples sont convertis (SC), puis sont mis en correspondance par le mapping ($\mathcal{M}_{d-a\ SC}$) à des assertions qui font référence à un même variant. En pratique, le variant ch6_18247207_c_A_G est instancié dans la BC une première fois. Puis la connaissance sur l'équivalence entre le variant initial 135411387_01 et le variant "pivot" ch6_18247207_c_A_G est ajouté à la BC :

$$135411387\_01 \doteq ch6\_18247207\_c\_A\_G$$

(ou en OWL : 135411387_01 `owl:sameAs` ch6_18247207_c_A_G)

Ensuite lors du traitement de variant rs1800460_01, celui-ci est converti (SC) et mis en correspondance ($\mathcal{M}_{d-a\ SC}$) à la liste d'assertion relatives, mais SNP-Converter vérifie dans la BC si le variant "pivot", ch6_18247207_c_A_G, lui correspondant est déjà représenté. Si c'est le cas il n'y est pas instancié à nouveau, et seule la connaissance sur leur équivalence est ajoutée :

$$rs1800460\_01 \doteq ch6\_18247207\_c\_A\_G$$

Ceci permet d'induire par un raisonnement basé sur la transitivité de l'opérateur $\doteq$ la connaissance suivante :

$$135411387\_01 \doteq rs1800460\_01$$

SNP-Converter a été utilisé dans le cadre d'une expérimentation d'intégration menée sur les variations génomiques spécifique au gène *LDLR*. Les Figures 3.14 et 3.15 illustrent les résultats obtenus. Trois jeux de données ont été soumis au SNP-Converter. Ceux-ci sont constitués tout d'abord de deux bases de données privées fournies par l'unité UMRS 538 de l'INSERM contenant 274 et 55 variants décrits

suivant deux formes non-conventionnelles. Ensuite, le troisième jeu de données est constitué des variants situés sur le gène *LDLR* contenu dans dbSNP en format XML (377). Parmi les 706 (274+55+377) variants différents utilisés pour peupler la BC, 634 sont considérés comme des individus uniques, *i.e.* représentés une seule fois dans la BC) et 35 autres sont représentés 2 ou 3 fois selon des représentations différentes au sein de la BC résultante. Ces derniers variants sont donc originellement contenus dans 2 ou 3 des jeux de données de départ.

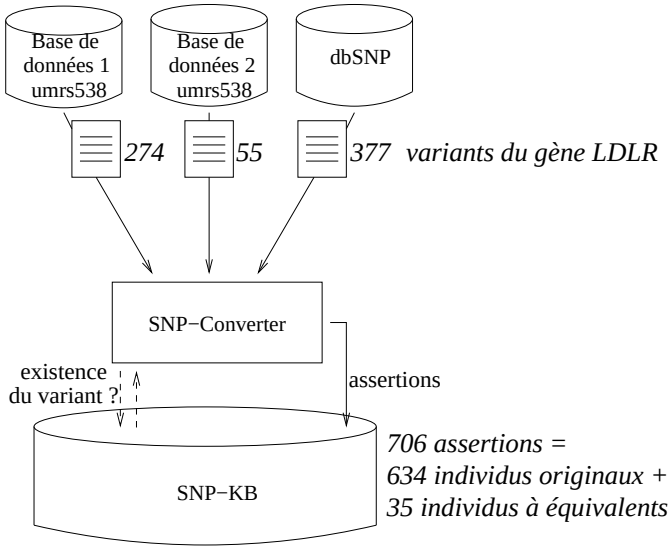

FIG. 3.14 – Utilisation du SNP-Converter comme wrapper et médiateur pour le peuplement d'une base de connaissances relative aux variations génétiques du gène *LDLR*

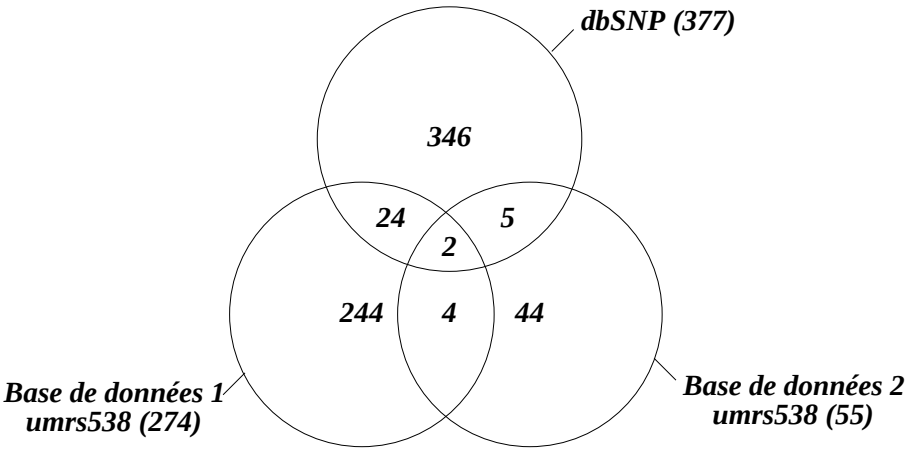

FIG. 3.15 – Diagramme de Venn représentant le recouvrement des trois jeux de données utilisées pour peupler la base de connaissances SNP-KB

3.2 Intégration de données pharmacogénomiques : iSO-Pharm

La pharmacogénomique est un domaine d'étude qui manipule des données complexes. La considération de sous-domaines (la pharmacologie, la biologie moléculaire, la médecine clinique, la génétique des populations, l'épigénomique entre autres) aux outils, objectifs, et sources de données distincts constitue un premier facteur de complexité. Les multiples niveaux de granularité entre ces sous-domaines, voire au sein d'un sous-domaine, sont également facteurs de complexité. Nous pouvons ajouter que ces données sont fréquemment interconnectées.

Ces différents facteurs de complexité justifient la construction d'un système d'intégration de données relatives à la pharmacogénomique. Une motivation supplémentaire vient du fossé existant entre d'un côté les données sur les connaissances pharmacogénomiques générales et de l'autre les observations cliniques qui ont permis de connaître ces relations. Un tel manque de relations peut être observé dans la base de données PharmGKB où coexistent, sans relation, des associations gène–médicament–maladie, et des jeux de données patients contenant des données relatives aux génotype, phéntoype et traitement de patients. Le même genre de lacune existe dans la base OMIM dont les entrées relatives aux maladies (survenant parfois dans le cadre d'un traitement) présentent une section "Clinical Synopsis" dont les données ne sont pas reliées aux variations génomiques associées, par exemple référencées dans dbSNP, voire même dans OMIM.

iSO-Pharm (pour *instanciate SO-Pharm !* en anglais) est un système qui intègre, selon la méthode proposée section 2 et dans le contexte d'une base de connaissances, des sources de données pharmacogénomiques relatives d'une part aux relations connues entre génotype–médicament–phénotype, et d'autre part à des données cliniques observées chez des patients. La Figure 3.16 représente l'architecture de ce système. Elle précise les sources de données intégrées, le rôle central de l'ontologie SO-Pharm et de mappings définis entre données (des sources) et assertions (associées à SO-Pharm). Il faut noter que chaque jeu de données de PharmGKB intégré nécessite la définition d'un mapping particulier de par le fait que chaque jeu est structuré suivant un schéma particulier.

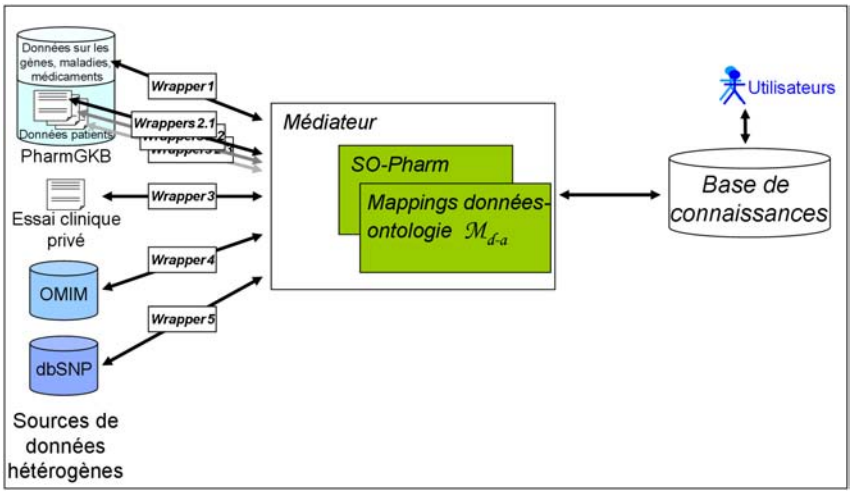

FIG. 3.16 – Architecture de iSO-Pharm instanciant l'architecture générale décrite Figure 3.10

L'objectif d'un tel système est d'intégrer à la fois des données cliniques (*i.e.* des données relatives au phénotype et au génotype de patients) et des données biologiques (*i.e.* enregistrées dans les bases de données biologiques et souvent issues d'expérience en biologie moléculaire) dans le cadre d'une BC de telle sorte qu'il soit possible d'utiliser la sémantique associée aux données pour l'extraction de connaissances en pharmacogénomique. Cet objectif est expérimenté dans la section 2.4 du chapitre 4. La base de

l'expérimentation décrite dans cette section est justement le peuplement d'une BC pharmacogénomique à partir

- de données cliniques de PharmGKB, consignées dans le cadre de l'investigation clinique des réponses de 61 patients asthmatiques à un médicament appelé le *montelukast* et
- de données biologiques de PharmGKB, dbSNP, OMIM, Gene[57], et KEGG Pathway[58].

Le peuplement de la BC associée à cette expérimentation mène notamment à la création de 61 assertions du concept "patient" (défini dans SO-Pharm), de 127 assertions du concept "clinical_item" ou de ses descendants, et des nombreuses assertions du rôle "presents_clinical_item" qui permet d'associer les instances des concepts "patient" et "clinical_item" conformément aux résultats de l'investigation clinique. Les données biologiques permettent de créer des assertions de concepts et de rôles relatives aux variations génomiques, aux gènes, aux médicaments, aux phénotypes, et à des réseaux métaboliques.

[57] http://www.ncbi.nlm.nih.gov/sites/entrez?db=gene
[58] http://www.genome.jp/kegg/pathway.html

4 Discussion

Il est possible de confronter deux façons de conceptualiser des connaissances : la première part des données pour définir des concepts, c'est en un sens le cas des méthodes semi-automatiques de construction d'ontologie, la seconde quant à elle part des concepts eux-mêmes auxquels des données peuvent être associées par la suite. Cette dernière est plus proche d'une construction manuelle dirigée par les connaissances d'un expert. Une mise en perspective proposée et illustrée dans l'introduction de la thèse de S. Rudolph fait le lien entre ces deux façons de conceptualiser le monde et les doctrines philosophiques inspirées des pensées d'Aristote et Platon, l'*empirisme* et le *rationalisme* [Rud06].

Un premier choix fort fait dans cette thèse est celui d'opter pour une construction manuelle des ontologies. Nous justifions celui-ci par deux arguments. Premièrement, les constructions d'ontologies que nous avons menées sont orientées vers deux objectifs précis qui sont l'intégration de données et la découverte de connaissances. Nous pensons que l'utilisation de méthodes de construction semi-automatiques à partir de données ou de textes introduisent un biais dans la représentation des connaissances trop contraignant vis à vis de nos objectifs. Notons que les bio-ontologies partagées sur les portails OBO-Foundry et BioPortal sont construites manuellement. Un second argument allant contre une construction semi-automatique est que les sources de données disponibles en pharmacogénomique ne recouvrent chacune que partiellement ce domaine en rapide évolution.

En revanche nous considérons l'utilisation des données et des textes comme le mode principal d'évaluation de la construction. En effet, le fait que les concepts et rôles d'une ontologie permettent la représentation des connaissances informelles contenues dans les bases de données et les textes est indispensable à l'accomplissement de nos deux objectifs (intégration de données et découverte de connaissances). C'est principalement par le test de cette capacité à représenter les connaissances que sont évaluées SNP-Ontology et SO-Pharm.

En ce qui concerne l'articulation des ontologies existantes avec les ontologies construites, nous avons aussi préféré définir manuellement la liste d'axiomes qui décrivent les relations entre concepts de différentes ontologies. Les raisons de ce choix sont similaires à celles qui motivent le choix d'une construction manuelle. Il est possible de considérer cette liste d'axiomes comme une *TBox* à part entière (*i.e.* une ontologie indépendante). Sur le plan théorique, c'est notamment ce qui semble le plus pertinent car cela évite d'avoir à incorporer des concepts externes dans l'ontologie, garantissant ainsi son intégrité (aussi bien que celle des ontologies articulées). Ceci évite également d'importer une ontologie volumineuse lorsque seules certaines branches de sa hiérarchie sont utiles. Cependant, cela impose une *modularisation* des ontologies qui n'est pas gérée actuellement par les outils standards utilisés pour la construction d'ontologies tels que OWL ou Protégé. La solution que nous avons adoptée est ainsi l'importation des ontologies articulées dans leur globalité comme l'autorisent ces outils. Associés à cette problématique, les travaux récents de Konev *et al.* proposent de définir des modules d'ontologies en LD qui pourraient être extraits et utilisés indépendemment [KLWW08]. Ceci permettrait d'utiliser les mécanismes de raisonnement sur SO-Pharm de façon moins contraignante.

Les propositions récentes concernant la représentation du temps en LD sont des progrès également profitables à la formalisation de domaines comme la pharmacogénomique, et par conséquent profitables aux futures versions de SO-Pharm [BGL08, LWZ08].

L'utilisation d'ontologies pour l'intégration de données est fréquemment discutée dans la littérature [CG05]. Notre motivation pour ce choix est ici renforcée par le fait que le second objectif de notre travail, qui fait suite à l'intégration de données, est l'étude de l'utilisation de connaissances formalisées pour guider la découverte de connaissances (présentée chapitre 4).

La méthode d'intégration que nous proposons peut être considérée comme un intermédiaire entre une approche médiateur et une approche entrepôt. En effet, elle est comparable à une approche entrepôt dans

le sens où les résultats de l'intégration sont matérialisés puis réutilisés notamment pour être analysés (voir chapitre 4). Néanmoins, nous la comparons également à l'approche médiateur pour souligner *(1)* le fait que les données manipulées restent dans les sources d'origine, ce qui est matérialisé est un ensemble d'assertions et *(2)* l'aspect dynamique de l'instanciation de l'ontologie qui s'effectue en réponse à une requête et permet de cette façon de créer des BC différentes en réponse à différentes requêtes.

Dans sa thèse soutenue en 2007, F. Saïs décrit une approche d'intégration sémantique fondée sur un *enrichissement sémantique* des données [Saï07]. Cet enrichissement consiste en l'ajout de termes décrivant les concepts et rôles d'une ontologie pour annoter les données. L'association donnée–terme est alors réalisée au travers d'un schéma XML spécifique. Le système d'intégration prend ensuite compte des annotations pour intégrer les données entre elles. Pour utiliser un vocabulaire comparable à celui de F. Saïs, nous pouvons dire, non pas que notre approche enrichit les données à l'aide de connaissances, mais qu'inversement nous enrichissons des connaissances à l'aide de données. Dans ce sens l'ajout dans une BC d'assertions de concepts et de rôles, préalablement associées à des données dans des mappings, peut être considéré comme un enrichissement de la BC.

Les contributions présentées dans ce chapitre ont donné lieu à deux publications. La première présente SNP-Converter et la seconde expose les méthodes associées à la construction et à la validation de l'ontologie SO-Pharm [CSTB+06, CSTND06].

Ce chapitre présente une approche d'intégration de données centrée sur une base de connaissances (BC) dont le résultat consiste en le peuplement de cette BC. L'objectif de cette intégration est non seulement d'intégrer des données issues de sources hétérogènes, mais également de leur associer une sémantique définie dans le cadre de la représentation des connaissances relatives à leur domaine. De fait, cette sémantique est utile pour intégrer les données, mais nous intéresse plus particulièrement pour améliorer les résultats d'un processus d'extraction de connaissances à partir des données intégrées. Nous abordons dans le chapitre suivant (chapitre 4) la façon dont une BC peut être associée à un processus d'ECBD pour en faciliter chacune des étapes, mais aussi comment elle peut être utilisée comme élément central d'un tel processus en appliquant les méthodes de fouille sur son contenu afin de bénéficier des connaissances aussi bien que des données disponibles.

Chapitre 4

Extraction de connaissances dans le contexte d'une Base de Connaissances

Nous avons défini et illustré la notion d'*Extraction de Connaissances guidée par les Connaissances du Domaine* (ECCD) dans le chapitre 2, section 4. Dans ce chapitre nous proposons et expérimentons une méthode d'ECCD dans laquelle les connaissances du domaine sont utilisées pour guider l'*étape de sélection* de données du processus (section 1). Ensuite, nous introduisons la notion nouvelle d'Extraction de Connaissances *à partir* d'une Base de Connaissances (ECBC) que nous proposons comme une approche particulière d'Extraction de Connaissances guidée par les Connaissances du Domaine (section 2). Nous présentons une méthode particulière d'ECBC appelée *Analyse des Assertions de Rôles* (AAR) qui explore les régularités existant dans les instanciations des rôles d'une BC en LD pour en extraire de nouvelles connaissances (2.3). Nous illustrons cette méthode par une expérimentation pour la découverte de connaissances en pharmacogénomique 2.4. Enfin la section 3 est une discussion de l'utilisation des connaissances pour la découverte de connaissances.

1 Sélection de données *guidée par* les connaissances du domaine

Dans cette section 1 nous proposons une méthode de sélection de données guidée par les connaissances. Celle-ci repose sur la définition de différents ensembles de mappings entre des sources de données et une BC. L'apport principal de cette méthode est de guider l'analyste dans la sélection de données à l'aide :

- de données disponibles dans un ensemble de sources de données hétérogènes connexes au domaine étudié,
- de l'expressivité des langages de LD et des mécanismes de raisonnement qui leur sont associés.

1.1 Motivation

Les méthodes de fouille de données, et plus particulièrement les méthodes symboliques, génèrent des résultats volumineux, redondants et complexes. Il en résulte une phase d'interprétation longue et fastidieuse qui limite le succès de l'approche.

Ce problème met en avant l'importance de la première étape de l'ECBD qu'est la préparation des données. De cette étape dépend la suite du processus depuis le paramétrage et le temps de calcul au succès même de l'extraction de connaissance utiles. Dans le cadre de la découverte de connaissances

en pharmacogénomique, Altman et Klein [AK02] soulignent l'importance du choix de sous-ensembles de données parmi la montagne de données post-génomiques disponibles. La réduction des données à prendre en compte pour la fouille a une influence directe sur le volume et la pertinence des résultats. Cette réduction est d'autant plus cruciale en biologie que les sources de données sont de plus en plus nombreuses et volumineuses [Bat08].

Réduire la quantité de données à fouiller par une opération de sélection permet de prendre en compte les connaissances (subjectives) des experts avant d'effectuer la fouille (voir chapitre 2 section 1.2.3). L'objectif d'une telle sélection est de réduire le temps de calcul et le volume des résultats produits sans en éliminer les éléments intéressants, ce qui facilite indirectement l'interprétation. Il s'agit donc d'éliminer progressivement et en accord avec les objectifs de la fouille les éléments redondants, triviaux, et dénués d'intérêt. Les connaissances alors utilisées le sont habituellement de façon manuelle à partir de connaissances propres à l'analyste et des informations qu'il peut collecter dans les bases de données.

L'utilisation de systèmes empiriques basés sur des méthodes statistiques et/ou d'apprentissage est un premier moyen d'assister l'analyste dans la sélection de données. Une vue d'ensemble de ces méthodes est proposée dans [SIL05], ainsi que dans la section 1 du chapitre 2.

Dans cette dernière section il est question d'un second moyen d'assister l'analyste par l'utilisation de connaissances, cependant les connaissances utilisées ne sont jamais représentées dans un formalisme qui permette la mise en œuvre de mécanismes de raisonnement. Ce qui nous intéresse ici est justement l'utilisation de connaissances dans le cadre de systèmes de sélection de données avec comme objectif de tirer parti des connaissances formalisées dans une BC en LD afin de guider à la fois le système (par des mécanismes de raisonnement) et celui qui le pilote. Le scénario d'extraction de connaissances en biologie exposé ci-après illustre la distinction entre le rôle des connaissances de l'expert, le contenu de bases de données, et l'utilisation d'une BC .

	variable_clin01	...	*variable_clin m*	*variant01*	*variant02*	...	*variant p*
patient01							
patient02							
...							
patient n							

TAB. 4.1 – Forme générale du jeu de données étudié dans le scénario

Scénario d'extraction de connaissances

Un biologiste étudie la pharmacogénomique liée au traitement de l'Hypercholestérolémie Familiale (HF) *à partir de données biologiques et génomiques pour un panel de patients traités. Le jeu de données dont il dispose présente pour chaque patient un ensemble de variables cliniques et plus de 500 génotypes de variants génomiques localisés sur différents gènes (Tableau* 4.1*).*

Pour sélectionner un sous-ensemble de données le biologiste peut utiliser :

sa propre connaissance *pour sélectionner les régions du génome où les variants sont susceptibles de l'intéresser : les gènes impliqués dans l'*HF *(LDLR, APOE, APOB, LPL) ; et plus particulièrement les exons, les promoteurs, et les régions flanquantes des exons de ces gènes. Cependant le biologiste est incapable, sur la base de sa seule connaissance, d'associer aux variants les régions sur lesquels ils sont situés.*

le contenu de bases de données *par exemple* Genome Browser *ou* dbSNP *lui permettent d'identifier parmi les variants explorés dans son panel, lesquels sont localisés dans les régions qui l'intéressent.*

l'utilisation d'une Base de Connaissances *lui permet potentiellement de savoir que les gènes en relation avec la pharmacogénomique de l'*HF *sont plus nombreux et incluent également les gènes MTTP et ESR1. Il peut alors sélectionner les variants localisés sur sa nouvelle liste de gènes sans passer par une base de données. Le biologiste peut également observer qu'il existe, au sein des variants, des sous-ensembles pertinents : les tag-SNP et les variants non-synonymes qu'il peut également isoler directement grâce à la* BC. *Il peut sélectionner les variants des gènes qui codent pour des protéines impliquées dans les réactions du métabolisme de l'atorvastatine*[59]*, ou plus généralement du métabolisme d'une statine (classe à laquelle appartient l'atorvastatine).*

Parce qu'elle intègre et structure les connaissances du domaine auquelle elle rattache les données brutes, qu'elle utilise un formalisme expressif et parce qu'elle peut être associée à des mécanismes de raisonnement, la BC est un outil précieux pour guider l'analyste dans un processus semi-automatique de sélection de données.

L'analyste, aussi expert soit-il, peut tirer parti de la représentation des connaissances encyclopédiques d'une ontologie pour orienter ses choix lors de la sélection. De plus la somme de connaissances disponibles laisse envisager que des tâches demandant moins d'expertise (comme par exemple la tâche de sélection, moins "pointue" que la tâche d'interprétation) puissent être réalisée par un analyste dont le niveau d'expertise est inférieur mais capable de s'appuyer sur le réferentiel déjà existant (*i.e.* la BC).

1.2 Méthode proposée

La méthode présentée ici a pour objectif, lors de l'étape de préparation dans un processus d'ECBD, d'aider l'analyste à sélectionner un sous-ensemble pertinent de données à fouiller que l'ensemble complet. Cette approche se veut indépendante de la suite du processus et notamment de la méthode de fouille utilisée.

Le principe est de permettre à l'analyste de faire cette sélection en prenant en compte les connaissances du domaine formalisées dans une BC préalablement développée. Pour cela, un mapping entre chaque base de données considérée et la BC doit être réalisé en collaboration avec un expert du domaine. La figure 4.1 décrit les quatre étapes principales de l'approche.

1 La première est l'*instanciation de la* BC. Celle-ci se fait suivant la méthode décrite dans le chapitre 3 section 2, *i.e.* sur la base de mappings définis entre les schémas de bases de données et l'ontologie. Ces mappings sont exploités par des *wrappers* qui instancient les concepts et rôles de l'ontologie à partir des tuples des bases de données considérées. Cette phase peut nécessiter diverses opérations de nettoyage et de transformation des données.

2 La deuxième étape consiste en la définition d'un *jeu de donnée initial*, ensemble de données extrait d'une ou plusieurs bases de données qui constite l'ensemble initial de données à analyser.

3 L'étape suivante est la définition d'un *mapping entre la* BC *et le jeu de donnée initial*. Ce mapping n'est pas défini manuellement mais est déduit des deux premières étapes. Son objectif est de permettre la répercussion d'une sélection d'individus dans la BC en une réduction en largeur (*i.e.* du nombre d'attributs) ou en longueur (*i.e.* des tuples) du jeu de données initial.

4 La dernière étape est la sélection par l'analyste d'un ensemble d'individus de la BC menant ainsi à la *réduction* du jeu de donnée initial en un *jeu de données réduit*. L'analyste ne sélectionne pas directement des données, mais des individus de la BC à l'aide du contenu des *TBox* et *ABox* . Il est ensuite possible, grâce au mapping précédent, de faire correspondre à la sélection d'individus une sélection de données.

[59] L'atorvastatine est un médicament de la classe des statines préscrit notamment pour prévenir la survenue d'accidents cardio-vasculaires.

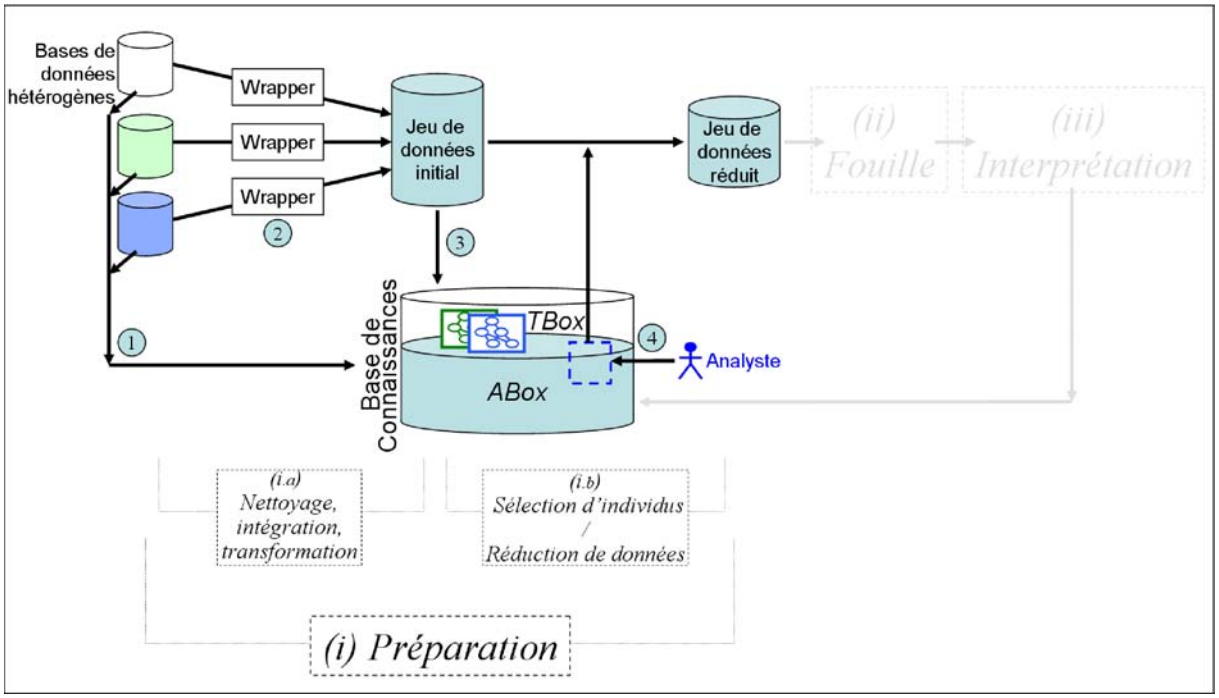

FIG. 4.1 – Description générale de la méthode de sélection de données guidée par les connaissances

Finalement, le jeu de données réduit peut être soumis aux étapes suivantes de l'ECBD : la fouille et l'interprétation. Les quatre étapes de la méthode de sélection proposée sont détaillées dans la suite de cette section. Celles-ci font notamment intervenir trois mappings positionnés Figure 4.2.

1.2.1 Instanciation de la BC

La première étape d'instanciation de la BC suit la méthode d'intégration guidée par une ontologie proposée chapitre 3 section 2.

Pour chaque base de données considérée en entrée du processus d'ECBD, un mapping entre le schéma de la base de données et les concepts, rôles et individus de l'ontologie, doit être défini par un expert du domaine. Ce mapping suit la définition 3.1 du mapping données-assertions $\mathcal{M}_{d-a}$ décrit dans le chapitre 3.

Il résulte de ce processus une BC instanciée à partir des données des bases de données considérées.

1.2.2 Définition du jeu de données initial

Le jeu de données initial est défini comme un ensemble de n-uplets suivant une relation n-aire unique $R_{init}(B_1, B_2, ..., B_m)$. Les attributs B_i de R_{init} peuvent être issus de différentes bases de données, c'est pourquoi la définition du jeu de données initial nécessite la définition d'un mapping entre les schémas des bases de données considérées pour l'ECBD et la relation du jeu de données initial.

Définition 4.1 (Mapping données – données) *Soit un quadruplet* $(\mathcal{S}, \mathcal{M}_{d-d}, H, R_{init})$ *où*

- $\mathcal{S}$ *est le schéma d'une base de données, i.e. un ensemble de relations n-aires de la forme* $R(A_1, A_2, ..., A_n)$ *et de domaine* $\prod_{i=1}^{n} D_i$ *tels que* A_i *est l'attribut d'indice i et de domaine* D_i.
- R_{init} *est la relation n-aire unique qui décrit le jeu de données initial* $R_{init}(B_1, B_2, ..., B_m)$,

– *$\mathcal{M}_{d-d}$ est une association entre les données de la base de données de schéma $\mathcal{S}$ et les données du jeu de donnée initial structurées selon son unique relation R_{init}.*

$$\Phi \rightsquigarrow \Upsilon \tag{4.1}$$

où Φ est une requête relationnelle arbitraire sur la base de données de schéma $\mathcal{S}$ et Υ est un ensemble d'insertions de tuples dans la relation unique R_{init} du jeu de données initial.

– *Enfin H un ensemble de fonctions de la forme $h_i(v)$ applicables aux différentes valeurs résultant des requêtes Φ pour les transformer dans le format de valeurs adéquat à leur insertion dans R_{init} décrit dans Υ.*

Exemple Soit deux relations $R_{clinique}$ et $R_{genetique}$ issues de deux bases de données dont on souhaite extraire une partie des données pour constituer le jeu de données initial qui suit la relation R_{init}

$R_{clinique}$ (Patient_id, Age, LDL_c, HDL_c, TG_c, xanthome, arc_corneen)
$R_{genetique}$ (Patient_id, rs28942078, rs28942079, rs28942080)

R_{init} (Patient_id, LDL_c_mgl, xanthome, rs28942076, rs28942078, rs28942079, rs28942080, rs28942081, rs28942082, rs28942083, rs28942084, rs28942085)

Deux exemples de mappings possibles $\mathcal{M}_{d-d}$ associent une requête SQL sur $R_{clinique}$ ou $R_{genetique}$ à une insertion dans R_{init} :

$\mathcal{M}_{d-d\ 1}$: SELECT Patient_id, LDL_c, xanthome FROM $R_{clinique}$ WHERE Age ≥ 18 $\rightsquigarrow$ INSERT INTO R_{init}(Patient_id, LDL_c_mgl, xanthome) VALUES (h_1(Patient_id), h_2(LDL_c), h_3(xanthome))

$\mathcal{M}_{d-d\ 2}$: SELECT Patient_id, rs28942078, rs28942079, rs28942080 FROM $R_{genetique}$ $\rightsquigarrow$ INSERT INTO R_{init}(Patient_id, rs28942078, rs28942079, rs28942080) VALUES (h_1(Patient_id), h_4(rs28942078), h_4(rs28942079), h_4(rs28942080))

Un tel mapping présente l'avantage de permettre de définir un jeu de données initial à partir de données issues de différentes bases et de permettre leur transformation. Le mapping $\mathcal{M}_{d-d\ 1}$ met en œuvre une transformation de données entre autres par l'utilisation de la fonction g' qui transforme les valeurs de concentration en LDL cholestérol circulant en $mol.l^{-1}$ *i.e.* l'attribut LDL_c, en valeurs de la même mesure mais avec une unité différente : le $mg.l^{-1}$ *i.e.* l'attribut LDL_c_mgl.

L'ensemble des mappings $\mathcal{M}_{d-d}$ impliquant les bases de données considérées dans le processus d'ECBD permet de constituer le jeu de données initial. Celui-ci peut être vu comme un tableau dont les colonnes seraient les attributs et les lignes seraient les n-uplets de la relation R_{init}. En suivant l'exemple précédent, ce tableau est conforme à la forme générale proposée dans le Tableau 4.1.

REMARQUE : Pour être tout à fait complet dans la description de la sélection de données, il faut noter l'existence d'une première étape de sélection, préalable à l'approche que nous décrivons. Celle-ci consiste dans le choix des bases de données à considérer pour l'ECBD. Nous ne la discutons pas dans ce travail.

1.2.3 Mapping entre la BC et le jeu de données initial

Les deux premiers mappings entre données et assertions, puis entre données et données permettent de déduire un mapping entre les données du jeu de données initial et les individus de l'ontologie. L'inversion de ce mapping fournit une correspondance entre certains individus de la BC et l'ensemble des attributs et n-uplets du jeu de données initial.

L'établissement du mapping entre la BC et le jeu de données s'appuie sur le fait que le jeu de données initial est constitué à partir de sous-ensembles de données qui ont servi à instancier la BC. D'une manière informelle, la déduction du mapping suit les phases suivantes : dans un premier temps le mapping données–assertions $\mathcal{M}_{d-a}$ général est réduit aux seules données du jeu de données initial ; ensuite, depuis le mapping réduit sont extraites des associations entre attributs du jeu de données et individus de la BC. Ces associations sont finalement inversées sous forme de relations entre individus et attributs. Si un individu est associé à l'attribut clé du jeu de données, l'association individu-attribut est étendue à l'ensemble du tuple.

Ces phases peuvent être formalisées selon les définitions suivantes.

Définition 4.2 (Mapping données – assertions indirect) *A partir des deux quadruplets* $(\mathcal{S}, \mathcal{M}_{d-a}, F, O)$ *et* $(\mathcal{S}, \mathcal{M}_{d-d}, H, R_{init})$ *suivant les définitions 3.1 et 4.1, nous définissons le quintuplet intermédiaire*

$$(R_{init}, \mathcal{M}_{d-a}, F, H, O)$$

où

- $\mathcal{M}_{d-a}$ *est l'association entre les données du jeu de données initial et un ensemble d'assertions de l'ontologie* O

  $$\Phi \rightsquigarrow \Psi$$

  où Φ *est une requête relationnelle arbitraire sur la relation* R_{init}*, et* Ψ *est un ensemble d'assertions de concepts et d'assertions de rôles de l'ontologie* O*.*
- *Enfin, un ensemble de fonctions composées à partir des ensembles* H *et* F *de la forme* $f_i(h_j^{-1}(v))$ *applicables aux différents types de valeurs* v *résultant des requêtes* Φ *sur le jeu de données initial, pour les transformer en noms d'individus dans* Ψ*.* $h_j(v)$ *est une fonction de transformation des valeurs* v *issues des bases de données considérées en leur format dans le jeu de donnée initial (voir définition 4.1).* $h_j^{-1}(v)$ *est l'inverse de cette fonction.* $f_i(v)$ *est une fonction de transformation des valeurs* v *réponses de* Φ *en noms d'individus (voir définition 3.1).*

Définition 4.3 (Mapping données – individus) *Soit* $(R_{init}, \mathcal{M}_{d-i}, F, H, O)$ *un autre quintuplet suivant la définition 4.2 avec* $\mathcal{M}_{d-i}$ *un mapping extrait de* $\mathcal{M}_{d-a}$*, qui est défini comme un ensemble d'associations 1 :n entre un attribut* B_i *de la relation* R_{init} *du jeu de données initial et un ou plusieurs individus* a_j *de* O*,*

$$B_i \rightsquigarrow \{\mathsf{a}_j\}$$

Les attributs B_i *peuvent être indifféremment des clés de la relation* R_{init} *ou non.*

La définition de ce mapping permet que chaque n-uplet (*i.e.* chaque clé) et que chaque attribut du jeu de données initial soit associé à un ou plusieurs individus de la BC.

Définition 4.4 (Mapping individus – données) *Selon la définition 4.3 du quintuplet* $(R_{init}, \mathcal{M}_{d-i}, F, H, O)$*, nous définissons le quintuplet* $(R_{init}, \mathcal{M}_{i-d}, F, H, O)$ *où*

– *$\mathcal{M}_{i-d}$, inverse de $\mathcal{M}_{d-i}$ ($\mathcal{M}_{d-i} = \mathcal{M}_{i-d}^{-1}$), est un ensemble d'associations binaires bijectives ($1:1$) entre un individu de la* BC *et un attribut B_i du jeu de données initial :*

$$a \rightsquigarrow B_i$$

B_i peut être une clé de la relation R_{init}.

Exemple Une partie du mapping $\mathcal{M}_{d-i}$ déduit entre R_{init} (voir l'exemple de la section 1.2.2) et l'ontologie *SNP-Ontology* est

$$\begin{array}{lcll} \texttt{Patient\_id} & \rightsquigarrow & f_1\,(h_1^{-1}(\texttt{Patient\_id})) & = \texttt{patient\_id} \\ \texttt{LDL\_c\_mgl} & \rightsquigarrow & f_2\,(h_2^{-1}(\texttt{LDL\_c\_mgl})) & = f_2 \circ\ h_2^{-1}(\texttt{LDL\_c\_mgl})) = \ f_2\,(\texttt{LDL\_c}) = \texttt{ldl\_c\_mol\_l} \\ \texttt{xanthome} & \rightsquigarrow & f_3\,(h_3^{-1}(\texttt{xanthome})) & = \texttt{xanthome} \\ \texttt{rs28942076} & \rightsquigarrow & f_4\,(h_4^{-1}(\texttt{rs28942076})) & = \texttt{rs28942076\_01} \end{array}$$

Si l'on observe le mapping proposé pour l'attribut `LDL_c`, il faut d'abord rappeler que l'attribut `LDL_c` de la relation $R_{clinique}$ avait été transformé par la fonction h_2 en `LDL_c_mgl` dans R_{init}. h_2^{-1} assure ainsi la première transformation inverse pour retrouver le format originel de l'attribut `LDL_c`. Ensuite la fonction g permet de transformer les valeurs de l'attribut en noms d'individus dans la BC, *i.e.* `ldl_c_mol_l`.

La partie correspondante du mapping inverse $\mathcal{M}_{i-d}$ entre individus et attributs est simplement

$$\begin{array}{lcll} \texttt{patient\_id} & \rightsquigarrow & \texttt{Patient\_id} & \\ \texttt{ldl\_c\_mol\_l} & \rightsquigarrow & \texttt{LDL\_c\_mgl} & = h_2 \circ\ f_2^{-1}(\texttt{ldl\_c\_mol\_l}) \\ \texttt{xanthome} & \rightsquigarrow & \texttt{xanthome} & \\ \texttt{rs28942076\_01} & \rightsquigarrow & \texttt{rs28942076} & \end{array}$$

La Figure 4.2 positionne les mappings $\mathcal{M}_{d-a}$, $\mathcal{M}_{d-d}$, $\mathcal{M}_{i-d}$ définis pour la sélection d'un jeu de données guidée par les connaissances du domaine ainsi que la forme des fonctions utilisées pour transformer les valeurs d'attributs en nom d'individus.

1.2.4 Sélection d'individus et réduction du jeu de données initial

La réduction du jeu de données initial repose sur une sélection, réalisée par l'analyste, d'individus de la BC. Pour cela l'analyste décrit un concept C_0 à partir des concepts et des rôles de l'ontologie. Le concept C_0 peut ainsi être explicitement défini dans l'ontologie ou correspondre à la description d'un nouveau concept (impliquant connecteurs logiques, concepts, rôles et individus) ou même le concept $\top$. Le mécanisme de raisonnement de *recherche d'instances* (*instance retrieval* en anglais) permet ensuite d'indiquer quels sont les individus instances de C_0.

Définition 4.5 ($\mathcal{A}_0$) *Soit $\mathcal{A}_0$ l'ensemble des individus a instances de C_0, tels que*

$$a \in \mathcal{A}_0 \textit{ si } \mathcal{O} \vDash C_0(a) \tag{4.2}$$

Les technologies du Web sémantique proposent différents langages de requête qui permettent de retrouver les individus instances d'un concept d'une ontologie comme par exemple SPARQL.

C'est lorsqu'il sélectionne ainsi des individus dans la BC que l'analyste peut bénéficier des connaissances formalisées dans l'ontologie.

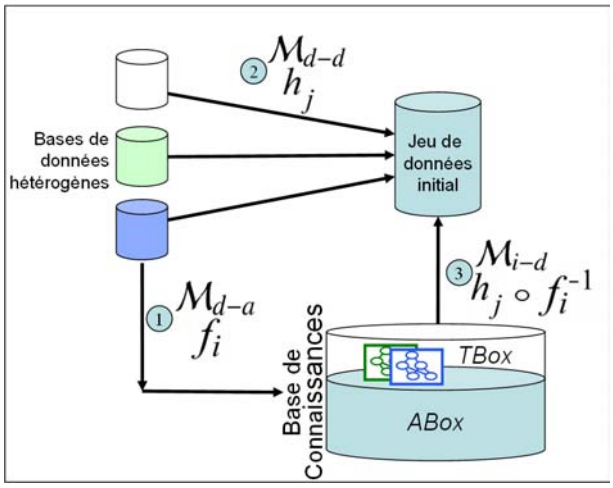

FIG. 4.2 – Positionnement et relations des trois mappings $\mathcal{M}_{d-a}$, $\mathcal{M}_{d-d}$, et $\mathcal{M}_{i-d}$. Les mappings $\mathcal{M}_{d-a}$ sont définis entre un schéma de bases de données et la Base de Connaissance. Les mapping $\mathcal{M}_{d-d}$ sont définis entre les schémas des bases de données et la relation du jeu de données intial. Le mapping $\mathcal{M}_{i-d}$ est déduit des deux précédents. Les fonctions symboliques associées aux mappings sont représentées. La forme générale des fonctions associées au mapping $\mathcal{M}_{i-d}$ est la composition de l'inverse de f_i et de h_j.

- L'analyste peut utiliser la hiérarchie de concepts pour sélectionner un ensemble d'individus instances d'un concept plus ou moins spécifique. La sélection progressive de concepts de plus en plus spécifiques au fur et à mesure des différentes itérations du processus permet de circonscrire un type d'individu afin d'en étudier les propriétés propres. A l'inverse il est possible de sélectionner des concepts de plus en plus généraux afin de valider la généralité d'une propriété mise en évidence sur un ensemble restreint d'individus.
- L'analyste peut utiliser les rôles et les restrictions associées pour sélectionner des individus présentant des propriétés particulières.

Une fois les individus sélectionnés, le jeu de données peut être réduit à l'aide de $\mathcal{A}_0$ et du mapping individu-données $\mathcal{M}_{i-d}$ selon les règles définies comme suit.

Définition 4.6 (Règles de réduction) *Soit B_i un attribut d'une relation R_{init}, a un individu d'une ontologie $\mathcal{O}$, le quintuplet $(R_{init}, \mathcal{M}_{i-d}, F, H, \mathcal{O})$ comme donné dans la définition 4.4, et un ensemble d'individus $\mathcal{A}_0$ instances d'un concept C_0. Si*

$$\begin{cases} \mathsf{a} \rightsquigarrow B_i & \in \mathcal{M}_{i-d}\ , \\ \mathsf{a} & \notin \mathcal{A}_0 \quad et \\ B_i & \text{clé de } R_{init} \end{cases} \tag{4.3}$$

alors le n-uplet dont B_i est la clé est supprimé du jeu de données initial.
De façon similaire, si

$$\begin{cases} \mathsf{a} \rightsquigarrow B_i & \in \mathcal{M}_{i-d}\ , \\ \mathsf{a} & \notin \mathcal{A}_0 \quad et \\ B_i & \text{non clé de } R_{init} \end{cases} \tag{4.4}$$

alors l'attribut B_i est supprimé du jeu de données initial.

En fonction du type d'individus sélectionnés, le jeu de données est réduit selon une dimension ou une autre.

Exemple Si le concept initial C_0 est défini par l'analyste comme suit :

$$C_0 \equiv \mathit{administrative\_item} \sqcup \mathit{phenotype\_item}$$

pour les quatres individus impliqués dans le mapping $\mathcal{M}_{i-d}$ proposé dans l'exemple précédent, les mécanismes de raisonnement sur SNP-Ontology donne les résultats suivant

$$\begin{aligned}&\text{SNP-Ontology} \models C_0(\texttt{patient\_id})\\ &\text{SNP-Ontology} \models C_0(\texttt{ldl\_c\_mol\_l})\\ &\text{SNP-Ontology} \models C_0(\texttt{xanthome})\\ &\text{SNP-Ontology} \not\models C_0(\texttt{rs28942076\_01})\end{aligned}$$

alors

$$\begin{aligned}\{\texttt{patient\_id, ldl\_c\_mol\_l, xanthome}\} &\in \mathcal{A}_0\\ \texttt{rs28942076\_01} &\notin \mathcal{A}_0\end{aligned}$$

et si l'on considère l'ensemble des $\mathcal{M}_{d-i}$, $\mathcal{A}_0$, et R_{init} qui détérmine les règles de réduction, seul l'individu `rs28942076_01` est inclu dans le mapping mais pas dans la sélection d'instance :

$$\begin{cases} \texttt{rs28942076\_01} \rightsquigarrow \texttt{rs28942076} & \in \mathcal{M}_{i-d}\ , \\ \texttt{rs28942076\_01} & \notin \mathcal{A}_0 \quad \textit{et} \\ \texttt{rs28942076} & \textit{non clé de } R_{init}. \end{cases}$$

En conséquence, l'attribut `rs28942076` de R_{init} est supprimé. En revanche, les attributs `Patient_id`, `LDL_c_mgl`, `xanthome` sont conservés pour constituer une nouvelle relation R_{reduit}. Les autres attributs de R_{init} relatif au génotype sont également supprimés de la relation du jeu de données initial. Au final, la transistion entre R_{init} et le schema R_{reduit} du jeu de donnée réduit est

R_{init} (`Patient_id`, `LDL_c_mgl`, `xanthome`, `rs28942076`, `rs28942078`, `rs28942079`, `rs28942080`, `rs28942081`, `rs28942082`, `rs28942083`, `rs28942084`, `rs28942085`)
↓
R_{reduit} (`Patient_id`, `LDL_c_mgl`, `xanthome`)

Les scénarios présentés dans la section 1.3 illustrent l'utilisation par un biologiste des connaissances du domaine pour réduire, en limitant la perte d'information, le nombre de n-uplets ou d'attributs dans le jeu de données initial.

1.3 Expérimentation pour la découverte de relations génotype–phénotype

1.3.1 Motivation

Nous présentons dans cette section des scénarios d'utilisation de notre approche de sélection de données guidée par les connaissances pour la recherche de *relations génotype–phénotype* introduites chapitre 1, section 1.3.

L'approche que nous proposons pour guider l'analyste dans sa sélection de données vise à s'appuyer, de façon semi-automatique, sur les connaissances disponibles du domaine. Ceci se justifie pleinement en biologie où de plus en plus d'ontologies sont construites et rendues disponibles sur Internet, comme sur les portails OBO Foundry et Bioportal évoqués chapitre 2, section 3.4.

Afin d'alléger la lecture, les mappings définis pour cette expérimentation et ayant donné lieu au développement de wrappers ne sont pas représentés. Cependant des exemples de ces mappings ont été proposés dans la section précédente (section 1.2).

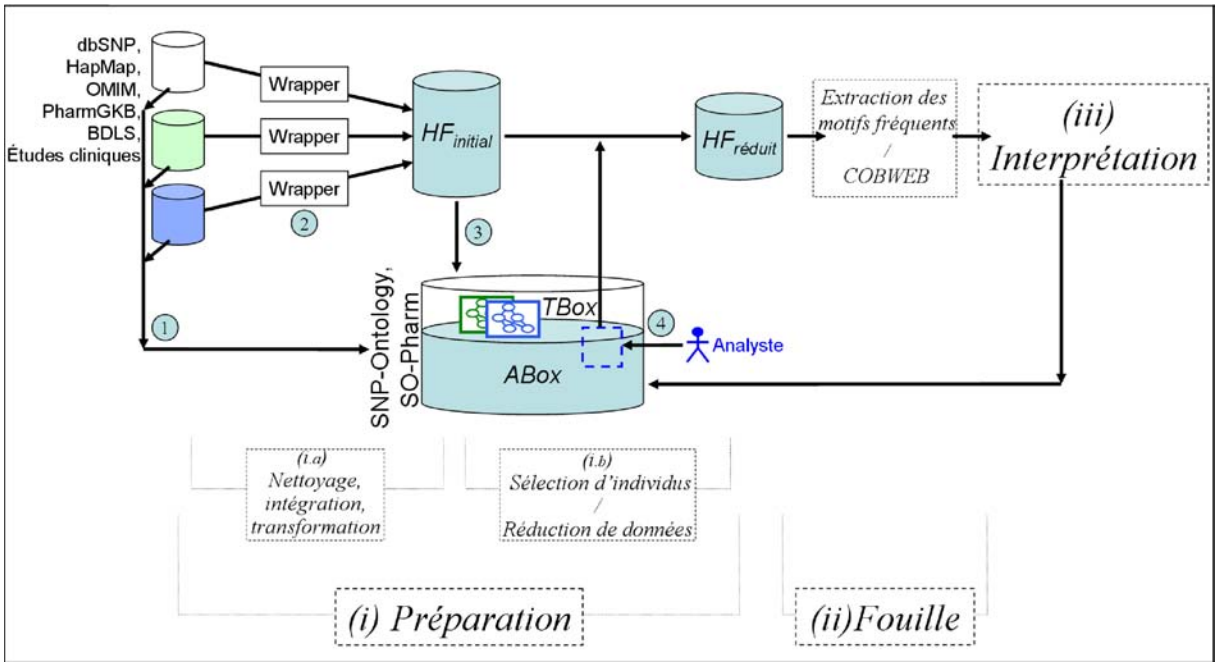

FIG. 4.3 – Approche pour la sélection de données (Figure 4.1) utilisée pour l'expérimentation *i.e.* la recherche de relations génotype–phénotype liées à l'HF

1.3.2 Hypercholesterolémie Familiale, sources de données et de connaissances

Notre expérimentation s'inscrit dans le cadre de la recherche de nouvelles connaissances relatives à l'hypercholestérolémie familiale (HF). L'HF est un désordre métabolique d'origine génétique, autosomal dominant causé par différentes mutations du gène *LDLR* [BDdG94]. Elle est caractérisée principalement par une augmentation importante de la concentration en cholestérol LDL (*Low Density Lipoprotein*) dans le sang.

L'objectif du processus d'ECBD mis en oeuvre ici est d'extraire des relations entre des *variants génomiques* (*i.e.* le génotype) et des *traits phénotypiques* (*i.e.* le phénotype). Des relations d'intérêt sont par exemple celles qui impliquent des variants génomiques *modulateurs*, *i.e.* un variant (ou un groupe de variants) qui a(ont) un effet modulateur sur la gravité de la pathologie étudiée (l'HF par exemple) ou sur un symptôme lié à celle-ci.

Par exemple, il existe différents niveaux de sévérité de l'HF qui sont fonctions de l'allèle observé pour deux variants génomiques du gène *APOE* (rs7412 et rs429358) [NBS$^+$06]. Ces variants modulateurs présentent un intérêt particulier en pharmacogénomique puisqu'ils sont souvent impliqués dans la modulation du métabolisme et de l'effet des médicaments [GBe07].

Les sources de données explorées dans le cadre de cette expérimentation ont deux origines différentes : *(i)* deux jeux de données privés, résultats d'investigations cliniques relatives à l'HF, *(ii)* des bases de données publiques (dbSNP, HapMap, OMIM, PharmGKB et des bases de données "Locus Spécifiques"[60]), dont certaines parties ont été utilisées pour instancier la SNP-Ontology. Cette instanciation suit l'approche décrite dans la section 1.2.1. La Figure 4.3 décrit la méthode (comme la Figure 4.1) dans le cas précis de la recherche de relations génotype–phénotype liées à l'HF.

De l'ensemble des sources de données considérées est extrait un jeu de données initial appelé $HF_{initial}$ constitué de 125 n-uplets correspondant à 125 patients impliqués dans une étude clinique liée à l'HF, et

[60] The WayStation, http ://www.centralmutations.org/

de 304 attributs relatifs au génotype (292/304) et au phénotype (12/304) des patients.

Le jeu de données $HF_{initial}$ implique :

(α) des patients atteints d'une hypercholestérolémie d'origine génétique (*i.e.* l'HF),

(β) des patients atteints d'une hypercholestérolémie d'origine non-génétique, et

(γ) des patients sains.

La majorité des attributs génétiques (289/293) provient du génotypage (*i.e.* les allèles portés) de chaque patient pour les variations génomiques du gène *LDLR* explorées. Un exemple d'attribut génétique de ce type sont les allèles observés pour la variation située à la position *Chr19 :11085058* (*e.g.* AA). Les attributs relatifs au phénotype décrivent les variables habituellement observées ou mesurées dans le cadre de l'exploration du métabolisme des lipides : par exemple, la concentration en cholestérol LDL circulant (*e.g.* $[LDL]_c$=3g.l^{-1}) ou la présence/absence de xanthome[61] chez le patient.

1.3.3 Méthodes de fouille

Pour évaluer la quantité de résultats de fouille de données dans le cadre de cette expérimentation nous utilisons deux méthodes de fouille de données :

– l'extraction des motifs fréquents présentée dans la section 1.3.2 du chapitre 2,

– la classification hiérarchique non supervisée COBWEB [Fis87].

La première méthode produit des motifs fréquents (MF) à partir desquels peuvent être isolés des sous ensembles de motifs non-redondants : les motifs fermés fréquents (MFF). Nous utilisons le nombre de motifs produits pour donner une estimation de la quantité de résultats à interpréter et le ratio du nombre de MF sur celui de MFF ($\frac{|MF|}{|MFF|}$) pour donner une estimation de la redondance des résulats.

COBWEB produit un ensemble de clusters organisés selon une hiérarchie. Le nombre de clusters nous sert d'indice pour juger de la complexité des résultats.

1.3.4 Sélection progressive de variants spécifiques – guidée par la subsomption

Le premier scénario s'appuie sur l'hypothèse que des relations génotype–phénotype pertinentes peuvent être plus facilement extraites d'un sous-ensemble de données ne contenant que les variants *codants*[62] ou les variants des *domaines protéiques conservés*[63]. Selon notre approche, ce genre de sélection résulte de la sélection dans l'ontologie SNP-Ontology des individus instances du concept le plus spécifique qui correspond au type de variant choisi. Cette sélection peut se faire par une navigation progressive dans la hiérarchie de l'ontologie en suivant les relations de subsomption. Le Tableau 4.2 illustre une sélection successive (au cours de différentes itérations du processus d'ECBD) des individus instances du concept *variant* puis de ses sous-concepts les plus spécifiques : successivement *variant* puis *coding_variant*, et enfin *conserved_domain_variant*. La sélection progressive d'un nombre décroissant d'individus se répercute en une diminution du nombre des attributs dans $HF_{initial}$: respectivement 289, 231, et 126 attributs. Les attributs relatifs aux variants non-codants sont éliminés dans un premier temps, puis sont écartés les variants codants localisés hors des régions correspondant aux domaines protéiques conservés.

En pratique la sélection de classes plus ou moins spécifiques en suivant l'organisation hiérarchique proposée par l'ontologie se fait par l'intermédiaire d'une partie d'un *plug-in* de Protégé 4 que nous avons développé et dont l'interface graphique est représentée en Annexe G.

Les conséquences que peut avoir la réduction du jeu de données sur la quantité et la significativité des résultats bruts de la fouille de données sont illustrées dans le Tableau 4.2. Pour cela, les différents jeux de données réduits obtenus ont été soumis aux deux méthodes de fouille de données utilisés pour évaluer

[61] Un xanthome est une tumeur bénigne cutanée souvent signe d'une anomalie des lipides.

[62] Localisés dans les régions codantes pour être précis.

[63] Dont les conséquences protéiques se localisent dans des domaines conservés pour être précis.

C_0	*variant*	*coding_variant*	*conserved_domain_variant*	*tag_snp*
Nombre d'attributs	289	231	126	198
MF (MFF)	6928 (255)	314 (24)	304 (12)	300 (28)
{ratio MF/MFF}	{27,17}	{13,08}	{25,33}	{10,71}
Clusters	194	186	56	40

TAB. 4.2 – Caractérisation quantitative des résultats bruts de fouille de données en fonction du nombre d'attribut sélectionnés

la quantité de résultats produits : l'extraction des motifs fréquents (avec l'algorithme Zart [SNK07]) et COBWEB (avec l'algorithme implanté dans Weka[64]). Quand tous les variants sont considérés (colonne *variant* du Tableau 4.2), le nombre total de motifs fréquents (MF) est de 6928 et le nombre de clusters de COBWEB est 194. Dans leur état brut, ces résultats de fouille sont complexes à interpréter. Le nombre de variables impliquées est important et il n'y a pas, excepté leur nom, d'informations contextuelles *a priori* qui permettent de les différencier. Par exemple, les variants codants ne peuvent pas être distingués des non-codants.

La quantité de résultats de fouille de données diminue progressivement lorsque moins d'individus donc moins d'attributs sont sélectionnés (colonnes *coding_variant* et *conserved_domain_variant*). Ainsi le nombre de MF passe de 6928 à 304 et le nombre de clusters de 194 à 56.

L'organisation hiérarchique, matérialisée par la relation de subsomption, est une des connaissances du domaine qui peut être utilisée pour réduire le volume du jeu de données à fouiller. Cependant, une telle sélection oblige à un compromis sur le type de variants à inclure dans l'étude.

1.3.5 Unification des variants à l'aide des Tag-SNP – guidée par les rôles et la compostion de rôles

Les résultats de la fouille du jeu de données $HF_{initial}$ présentent un proportion importante de MF triviaux ou redondants. Ceci est dû en partie au fait que certains variants du jeu de données appartiennent aux mêmes haplotypes. Comme décrit dans la section 2.4 du chapitre 1, un haplotype désigne un groupe de variants transmis conjointement et de façon homogène à travers les générations. Il est possible d'identifier, au sein des haplotypes, un ensemble minimal de variants appelées *Tag-SNP* dont l'observation suffit à prédire l'allèle présenté par les autres variants de l'haplotype. Réduire un ensemble de variants membres d'un haplotype à ses tag-SNP permet de réduire les relations qui traduisent la dépendance entre ces variants, et ainsi réduit la redondance des résultats.

La Figure 4.4 montre un haplotype et sa représentation dans l'ontologie SNP-Ontology. Cet haplotype est composé des variants rs_001, rs_002, rs_003, et rs_004, et peut être remplacé par son unique tag-SNP rs_004. La description d'un haplotype (ici le NA01234) met en lumière l'existence d'une dépendance fonctionnelle entre un (ou plusieurs) tag-SNP (rs_004) et les autres membres de l'haplotype (rs_001, rs_002, rs_003). Cette dépendance est représentée dans la SNP-Ontology comme suit

$$\{rs\_001, rs\_002, rs\_003\} := \exists\, isHaplotypeMemberOf\ \{haplotype\_NA01234\} \sqcap \exists\, isHaplotypeMemberOf \circ isTaggedBy\ \{rs\_004\}$$

[64] http ://www.cs.waikato.ac.nz/ml/weka/

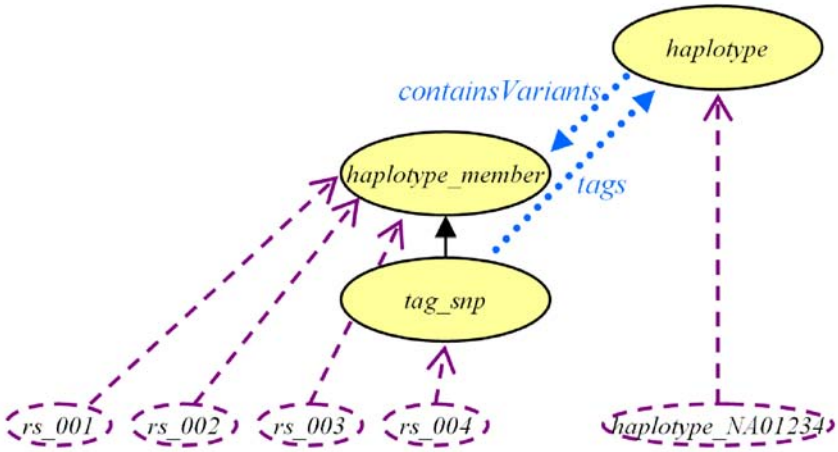

FIG. 4.4 – Concepts de SNP-Ontology instanciés par des individus représentant des variations génomiques (rs_001, rs_002, rs_003, et rs_004) et un haplotype (NA_01234). *Légende* : les ovales pleins sont des concepts, les ovales en tirets sont des individus, la ligne pleine est une relation de subsomption, les lignes en tirets ronds sont des rôles, les lignes en tirets plats sont des assertions.

et inversement

$$\begin{aligned}\{rs\_004\} := \quad & \exists\, tags\, \{haplotype\_NA01234\} \sqcap \\ & \exists\, tags \circ containsVariants\, \{rs\_001\} \sqcap \\ & \exists\, tags \circ containsVariants\, \{rs\_002\} \sqcap \\ & \exists\, tags \circ containsVariants\, \{rs\_003\}\end{aligned}$$

L'ontologie contient l'ensemble des descriptions des haplotypes des gènes génotypés pour la population étudiée dans l'étude $HF_{initial}$. Les connaissances sur les haplotypes sont intégrées à l'ontologie lors de la phase d'instanciation de notre approche, à partir des données du projet HapMap, et de données issues du logiciel Haploview [Con03, BFMD05].

Le concept de tag-SNP est explicitement décrit dans SNP-Ontology de la façon suivante :

$$tag\_snp \equiv \exists\, tags \circ containsVariant.Variant \tag{4.5}$$

Ainsi, en limitant la définition du concept C_0 à la définition des tag-SNP (*i.e.* $C_0 \equiv tag\_snp$), l'analyste entraîne la suppression au sein du jeu de données, des attributs qui ne concernent pas des tag-SNP. Suivant notre exemple basé sur l'haplotype NA01234, ceci revient finalement à la suppression des colonnes rs_001, rs_002, et rs_003 du jeu de donnée $HF_{initial}$. Au niveau de l'ensemble du jeu de données $HF_{initial}$, le Tableau 4.2 montre qu'une telle sélection réduit le nombre d'attributs (289 à 198) et diminue considérablement la quantité de résultats produits par les deux méthodes de fouille. La réduction des résultats de fouille est due premièrement à la réduction du nombre d'attributs et deuxièmement à la réduction du nombre d'associations liées à la dépendance fonctionnelle (*i.e.* la co-segrégation) entre les variants d'un même haplotype. Le ratio $|\text{MF}|/|\text{MFF}|$ donne une idée de la redondance[65] qui existe au sein des motifs extraits lors de l'extraction de motifs fréquents et indique ainsi que la redondance entre les MF diminue lorsque le jeu de données est réduit en utilisant les tag-SNP.

REMARQUE : Les haplotypes sont des constructions statistiques dont la précision est dépendante de l'échantillon d'individus utilisé. La réduction du jeu de données sur la base de la composition des haplotypes souffre donc du même biais.

[65] Un motif est d'autant plus redondant qu'il retrouvé comme sous-motif d'un grand nombre d'autres motifs.

1.3.6 Sélection de patients – guidée par la définition de concepts

Les deux premiers scénarios visaient à réduire le nombre d'attributs (relatifs au génotypage de variants génomiques). Le troisième scénario, décrit dans cette section, illustre quant à lui la réduction du nombre de n-uplets (*i.e.* de patients) du jeu de données $HF_{initial}$. Pour ce faire, l'analyste sélectionne des individus instances des concepts décrits, non plus dans l'ontologie SNP-Ontology, mais décrit dans SO-Pharm dont la SNP-Ontology ne constitue qu'une partie (voir chapitre 3, section 1.3).

Les concepts, rôles, et individus de SO-Pharm permettent de décrire de nouveaux concepts qui peuvent présenter un intérêt particulier dans le cadre de l'exploration de l'HF. Le jeu de données regroupe notamment trois classes différentes de patients (α, β, et γ), qui ne sont pas initialement représentées dans SO-Pharm, mais qu'il est intéressant de regrouper dans le cadre de l'étude afin d'en explorer les propriétés caractéristiques et discriminantes. Pour cela, l'analyste peut utiliser SO-Pharm et les individus créés lors de l'étape d'instanciation pour définir le concept C_0 correspondant à la classe de patients qu'il veut étudier :

$$\begin{aligned}
\text{patients } \alpha \ : \mathsf{C}_0 \ \equiv \ & \textit{patient} \sqcap \\
& \exists\, \textit{hasGenotypeItem}\ \{\textit{LDLR_mutation}\}
\end{aligned}$$

$$\begin{aligned}
\text{patients } \beta \ : \mathsf{C}_0 \ \equiv \ & \textit{patient} \sqcap \\
& \exists\, \textit{hasGenotypeItem}\ \{\textit{no_LDLR_mutation}\} \sqcap \\
& \exists\, \textit{hasPhenotypeItem}\ \{\textit{high_LDL_in_blood}\}
\end{aligned}$$

$$\begin{aligned}
\text{patients } \gamma \ : \mathsf{C}_0 \ \equiv \ & \textit{patient} \sqcap \\
& \exists\, \textit{hasGenotypeItem}\ \{\textit{no_LDLR_mutation}\} \sqcap \\
& \exists\, \textit{hasPhenotypeItem}\ \{\textit{normal_LDL_in_blood}\}
\end{aligned}$$

L'utilisation du mécanisme de recherche d'instances permet de déterminer quelles sont les instances du concept C_0. Selon l'approche décrite, cela a se répercute au niveau des données, qui vont être réduites à un sous-ensemble de n-uplets qui partagent un attribut en commun ou qui appartiennent à une même classe de patients. L'intérêt principal de cette réduction est qu'elle peut se faire à l'aide d'attributs ou de classes qui ne sont pas présents dans le jeu de données initial $HF_{initial}$ mais qui sont représentées dans l'ontologie SO-Pharm.

En pratique, la définition de C_0 s'effectue de la même manière que dans le premier scénario, grâce à l'utilisation d'un *plug-in* de Protégé 4 (voir Annexe G).

1.4 Bilan

Nous avons présenté dans cette section une méthode de sélection de données qui, moyennant la définition par l'analyste d'un ensemble de mappings adéquats, lui permet de bénéficier du contenu de la BC pour réduire intelligemment un jeu de donnée initial avant la fouille.

La proposition décrite dans cette section pour guider la sélection de données à l'aide des connaissances du domaine et son illustration par des scénarios de recherche de relations génotype–phénotype ont été publiées dans le journal *BMC Bioinformatics* [CSTB+08].

Dans l'idée d'aller plus loin dans l'utilisation des connaissances disponibles pour l'extraction de connaissances, la section suivante présente une approche intégrée d'Extraction de Connaissance *à partir* de Base de Connaissance (ECBC) où l'ensemble du processus d'ECBD est revisité en présence d'une BC. Cette approche présente en outre l'avantage d'alléger le travail de l'analyste en n'exigeant que la définition

des mappings données–assertions ($\mathcal{M}_{d-a}$) nécessaires au peuplement de la BC à partir d'un ensemble de bases de données hétérogènes.

2 Extraction de Connaissances *à partir* d'une Base de Connaissances – ECBC

Nous proposons une approche particulière d'Extraction de Connaissances *guidée par* les Connaissances du Domaine (ECCD) appellée l'Extraction de Connaisances *à partir* d'une Base de Connaissances (ECBC). La nouveauté de celle-ci est que la BC n'est plus positionnée en marge du processus mais est l'élément central dont sont à la fois extraits les éléments à fouiller et les connaissances pour guider la fouille.

2.1 Description générale

Nous proposons une approche d'ECCD dont l'originalité principale est de travailler à partir des *TBox* et *ABox* d'une BC. L'hypothèse sous-jacente est l'existence de régularités porteuses de connaissances nouvelles et significatives dans l'instanciation (définie et induite) d'une BC.

Il s'agit donc d'appliquer des méthodes de fouille de données sur un ensemble d'assertions de la BC dans le but de déceler des régularités interprétables sous forme de connaissances pertinentes qui raffineront la BC. Nous appelons cette approche l'Extraction de Connaissances *à partir* d'une Base de Connaissance (ECBC) par distinction avec l'Extraction de Connaissances *à partir* de Bases de Données (ECBD).

Deux obstacles se posent à la mise en œuvre d'une telle approche :

- premièrement les BC ne contiennent souvent qu'une quantité de connaissances restreinte comparé au contenu de bases de données ou de corpus de textes ;
- deuxièmement les algorithmes de fouille de données sont développés pour manipuler des données et non des assertions, de plus les résultats de ces algorithmes ne sont pas représentés suivant un formalisme de représentation des connaissances.

Nous proposons de dépasser la première limite en développant des mappings entre le contenu des bases de données du domaine et l'ontologie (ou *TBox*) *(0)*. Ces mappings serviront de base à des *wrappers* développés spécialement pour peupler l'ontologie à partir du contenu de bases de données.

Pour surmonter la deuxième limite, il est nécessaire de réaliser une étape de transformation *(i)* des assertions de l'ontologie en un format compatible avec le format d'entrée de la méthode de fouille choisie. Après l'étape de fouille proprement dite *(ii)*, il est également nécessaire de réaliser une étape de transformation inverse *(iii)* des résultats de fouille en axiomes et assertions dans le formalisme de l'ontologie.

Notre méthode se divise ainsi en 4 étapes principales (*0, i, ii, iii*) dont les 3 dernières peuvent être comparées aux trois étapes principales du processus d'ECBD : *(i)* la préparation des données, *(ii)* la fouille, et *(iii)* l'interprétation. Nous supposons ici que la *TBox* de la BC est déjà construite. La Figure 4.5 représente schématiquement cette approche *itérative* et *interactive*.

2.2 Application conjointe des Logiques de Descriptions et de l'Analyse de Concepts Formels dans le contexte de l'ECBC

L'existant le plus proche de la méthode d'ECBC proposée ci-dessus vient de travaux qui font intervenir conjointement des BC formalisées en LD et des méthodes d'Analyse de Concepts Formels (ACF) (chapitre 2, section 1.3.1). LD et ACF partagent, malgré des différences fondamentales, deux principes : la notion de *concept* et l'*organisation hiérarchique* de ces concepts. Bien que différente en LD et en ACF, la notion de concept repose sur la même idée fondamentale de collection d'objets partageant un certain nombre de propriétés. Aussi l'organisation en hiérarchie des concepts formels produite par l'ACF présente des similitudes avec l'organisation des concepts d'une ontologie en LD. Ces similitudes rendent possible

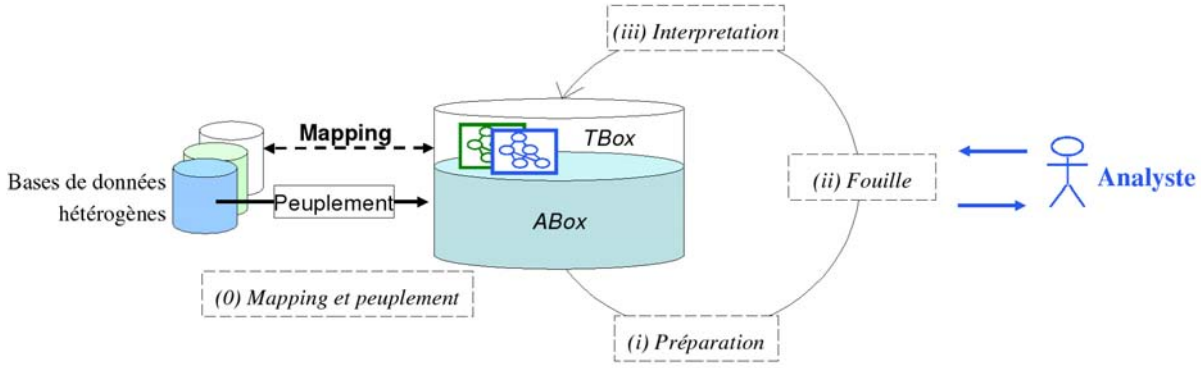

FIG. 4.5 – L'Extraction de Connaissances *à partir* d'une Base de Connaissances ou ECBC

l'utilisation conjointe de méthodes ou outils provenant des deux domaines. Toutefois, les différences principales entre concepts en LD et concepts formels sont, premièrement la façon dont ils sont obtenus et deuxièmement la façon de les décrire.

En LD les *concepts* sont obtenus de façon manuelle ou semi-automatique, par un expert du domaine étudié, dans l'objectif de formaliser les concepts d'intérêt du domaine en question. La description d'un concept en LD est construite à partir de concepts atomiques (des prédicats unaires), de rôles atomiques (des prédicats binaires), et des constructeurs de concepts fournis par le langage de LD utilisé ($\exists, \forall$ par exemple). Les concepts décrits, ainsi que les rôles, servent dans un second temps à la spécification des propriétés des objets. Pour plus de détails voir la section 2.2 du chapitre 2 ou [BCM+03].

En ACF les *concepts formels* sont obtenus à partir de contextes formels, qui spécifient les attributs (ou propriété) présentés par chaque objet. Dans un tel contexte, un concept formel est décrit par un ensemble d'objets (son extension) et un ensemble d'attributs (son intension) de telle sorte que l'intension contienne exactement l'ensemble des attributs que les objets de l'extension ont en commun et qu'inversement, l'extension contienne exactement l'ensemble des objets qui partagent tous les attributs de l'intension. Pour plus de détails voir la section 1.3.1 du chapitre 2 ou [GW99].

Dans les deux cas les descriptions associées aux concepts permettent de les organiser en une hiérarchie. Toutefois les deux types de concept, décrits de façons distinctes, produisent deux types de hiérarchies distinctes.

REMARQUE : Certains auteurs utilisent également les notions d'intension et d'extension concernant les concepts de LD. Dans ce cas, l'intension du concept est la description du concept et l'extension est l'ensemble des individus (*i.e.* des objets) instances du concept en question.

Baader *et al.* utilisent l'ACF pour construire en partant du bas une hiérarchie de concepts à partir d'un ensemble fini de concepts $\{C_1, ..., C_n\}$ [BS04]. Le principe de la méthode repose sur la définition d'un contexte formel à partir de l'ensemble de concepts de départ et soit de leurs conjonctions, soit de leurs subsumants communs les plus spécifiques. La méthode d'ACF utilisée sur le contexte est l'algorithme d'*exploration d'attributs* de Ganter [Gan84] qui permet de déterminer des *implications* de la forme $B_1 \rightarrow B_2$. Les implications permettent de dériver des relations de subsomption entre les concepts

de départ et leurs conjonctions (par exemple $B_1 \sqsubseteq B_2$) ; ou entre les subsumants communs les plus spécifiques des concepts de départ et les concepts de départ, de sorte à générer progressivement et de bas en haut une hiérarchie. En pratique ce travail est très peu exploitable, car les hiérarchies générées automatiquement sont volumineuses en raison du fait que tous les concepts qu'il est possible de décrire à partir des concepts de départ sont construits et inclus dans la hiérarchie. De plus la méthode s'appuie sur l'hypothèse forte qu'un subsumant commun le plus spécifique existe toujours et peut toujours être trouvé, ce qui n'est pas le cas en pratique. Enfin ce travail prend en compte la *TBox* sans exploiter les connaissances de la *ABox*.

Des résultats plus exploitables dans le cadre de l'ECBC résultent de deux travaux qui se complètent : l'*Exploration Relationnelle* (que nous noterons ER)[Rud06] et la proposition de complétion des BC en LD par Baader *et al.* [BGSS07].

L'Exploration Relationnelle (ER) décrite par Rudolph s'appuie sur une extension de l'algorithme d'exploration d'attributs dans un contexte de LD. Pour cela l'ER se base sur la définition du contexte lié à l'interprétation $\mathcal{I}$ en LD

Définition 4.7 (Contexte - LD) *Soit $\mathcal{I}$ une interprétation sur le domaine Δ, M un ensemble de description de concepts de ce domaine en* LD*, et I_c une relation d'incidence. Le contexte $\mathcal{K}_{\mathcal{I}}(M)$ lié à l'interprétation $\mathcal{I}$ est défini par le triplet (Δ, M, I_c), où quand la relation d'incidence I_c associe à un individu δ de Δ une description de concept C de M alors l'individu δ est instance du concept $C^{\mathcal{I}}$. Plus formellement,*

$$\delta I_c C^{\mathcal{I}} \Leftrightarrow \delta \in C^{\mathcal{I}}.$$

A partir de cette définition il est démontré que les implications extraites de $\mathcal{K}_{\mathcal{I}}$ par l'exploration des attributs coïncident avec certains axiomes valides selon $\mathcal{I}$ (voir [Rud06]). Ainsi si $C, D \subseteq M$ alors l'implication $C \rightarrow D$ est extraite de $\mathcal{K}_{\mathcal{I}}$ si et seulement si $\mathcal{I}$ satisfait l'axiome $C \sqsubseteq D$. L'ER permet d'explorer les axiomes d'inclusion par cette correspondance et de vérifier leur validité dans le domaine (selon $\mathcal{I}$) à travers un système de questions-réponses à un expert du domaine. Si l'assertion proposée n'est pas explicitement décrite dans la *TBox* et ne peut pas être induite par le mécanisme de raisonnement de subsomption, alors l'expert est interrogé sur sa validité. Si l'assertion est vraie selon l'expert, elle vient enrichir la *TBox* . Si elle est fausse, l'expert doit fournir un contre exemple qui sera ajouté à la *ABox* de la BC. De cette façon l'implication ne sera plus extraite lors d'une exploration suivante et la BC (*TBox* et *ABox*) est progressivement raffinée.

La complétion des BC en LD proposée par Baader *et al.* [BGSS07] propose des améliorations permettant la mise en oeuvre effective de l'ER. Premièrement, elle formalise l'utilisation de l'ACF sur des contextes partiels. Cette utilisation est nécessaire à la prise en considération d'objets partiellement décrits par les méthodes de ACF, comme l'exploration d'attributs. Suivant l'hypothèse du monde ouvert (détaillée chapitre 2 section 2.2) les individus d'une BC en LD sont justement des objets partiellement décrits. Deuxièmement, la méthode limite à la seule subsomption les constructeurs logiques autorisés dans les descriptions de concepts considérés par la contexte (*i.e.* les concepts de M de $\mathcal{K}_{\mathcal{I}}$). Ceci permet de réduire le nombre d'implications, et donc de questions posées à l'expert.

Le bénéfice commun des résultats de ces deux travaux est illustré par une méthode d'acquisition semi-automatique d'axiomes en LD à partir de corpus de textes dans [VR08].

Une première limite des méthodes basées sur l'Exploration d'Attributs est de n'exploiter que les implications du contexte, *i.e.* les règles dont la confiance est égale à 1. C'est justement ce qui permet d'exclure un axiome $C \sqsubseteq D$ lorsque l'expert donne un contre-exemple à un axiome, cela revient à

ajouter un objet au contexte qui présente la propriété C sans la D ou inversement. Ce nouvel objet rend forcément la confiance de la règle $C \rightarrow D$ inférieure à 1, ce qui évite l'implication entre C et D. Nous pensons que cette limite est trop forte et peut empêcher la mise en évidence de concepts intéressants à inclure dans la *TBox*. Quand une BC est peuplée de nombreux individus, quelque soit le mode utilisé pour son peuplement (manuel ou automatique), elle reste une représentation d'une réalité particulière soumise aux nombreux artéfacts que cela implique : par exemple le biais dans la représentation des connaissances, la reproduction ou l'introduction de bruit, d'erreurs lors du peuplement de la BC, la difficulté à prendre en considération les cas extrêmes.

De plus selon la configuration de la BC (et notamment de son peuplement), le nombre d'implications et donc de questions posées à l'expert peut être très élevé sans que celles-ci n'apportent aucun bénéfice dans la représentation des connaissances qui intéressent l'expert. Par exemple, un clinicien qui explore une BC représentant les patients d'un hôpital, leurs dossiers médicaux et administratifs, peut selon la façon avec laquelle a été peuplé la BC, générer de nombreuses implications évoquant des connaissances d'ordre administratif ("chômeur" → "assuréCMU" ou "transportEnAmbulance" → "ActeDeRadiographie") et finalement très peu de connaissances d'ordre médical qui puissent l'intéresser.

Nous proposons dans la section suivante une méthode d'ECBC qui utilise la complémentarité des LD et de l'ACF commme Rudolph *et al.* et Baader *et al.* . Notre méthode se distingue notamment par :

- la transcription des connaissances en données accessibles à la fouille,
- la méthode de fouille utilisée,
- la position de l'analyste,

et s'oriente plus particulièrement vers une mise en application opérationnelle sur des données réélles.

2.3 Analyse des Assertions de Rôles – AAR

L'Analyse des Assertions de Rôles – où AAR – est une approche particulière d'Extraction de Connaissances *à partir* de Bases de Connaissances (ECBD). L'AAR explore les régularités dans les relations directes et indirectes entre instances d'une BC en LD, *i.e.* les régularités des assertions de rôles et de leur composition. La section 2.3.1 décrit l'AAR d'un point de vue général, puis la section 2.3.2 la détaille étape par étape. Enfin la section 2.4 présente des résultats expérimentaux obtenues en pharmacogénomique par AAR.

2.3.1 Description générale

L'AAR s'attache à analyser les régularités présentes dans la *ABox* (*i.e.* les *assertions de concepts* et *de rôles*) d'une ontologie en LD en utilisant les méthodes d'Analyse de Concept Formel (ACF) et d'extraction de *Règles Minimales Non-Redondantes Réduites* ($\mathcal{RMNR}$). Ces régularités sont susceptibles de refléter l'existence de connaissances implicites dans la BC et de mettre en lumière des relations *intéressantes* (selon l'analyste) mais masquées qui prennent la forme de relations *indirectes* ou *complexes* entre les individus de la BC. Une relation est indirecte si sa représentation nécessite l'enchaînement de plusieurs rôles ; une relation est complexe si elle implique des relations vers plusieurs individus distincts.

Pour cela nous proposons d'utiliser, dans le cadre d'un processus semi-automatique et itératif, le formalisme des LD pour définir des attributs analysés par ACF ; l'exploration par ACF nous permet de son côté d'obtenir, ou d'affiner, des descriptions en LD. De façon informelle, les LD exploitent les résultats obtenus par ACF pour acquérir interactivement des connaissances, et l'ACF bénéficie des LD pour exprimer des *connaissances relationnelles i.e.* des connaissances sur les relations entre individus [Rud06].

Le prérequis indispensable à une telle approche est évidemment de disposer d'une ontologie en LD instanciée pour pouvoir en utiliser les assertions. Ensuite l'AAR se décompose schématiquement en trois

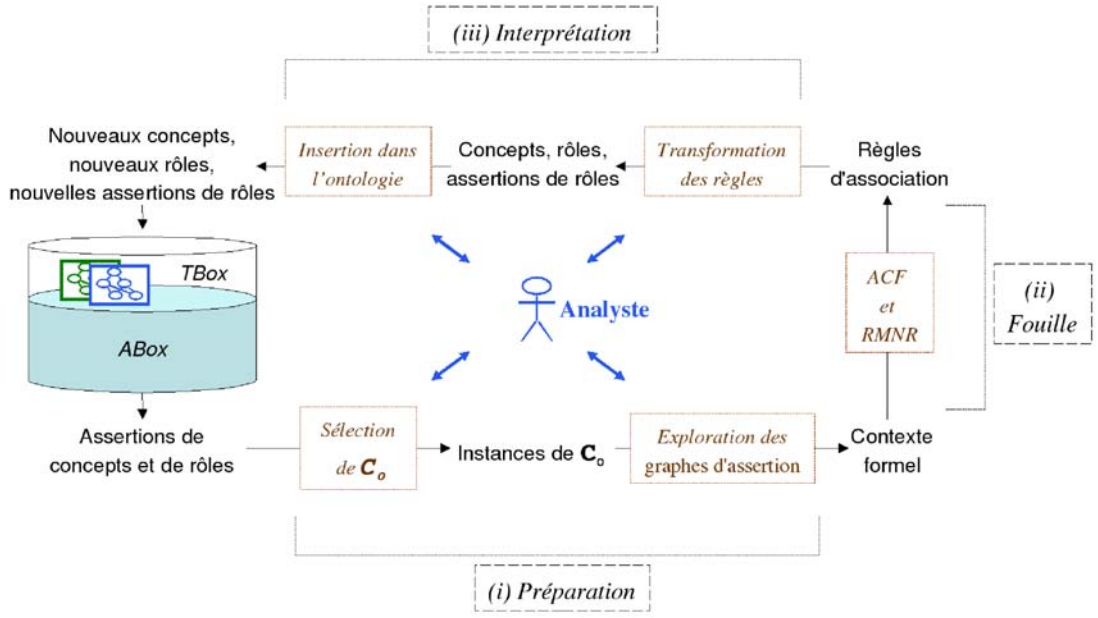

FIG. 4.6 – L'Analyse des Assertions de Rôles (AAR) et des ses différentes étapes

parties principales.

- La première partie (Figure 4.6, *(i) Préparation*) vise à transformer les assertions en un contexte formel, format de données compatible avec les méthodes d'ACF et d'extraction des $\mathcal{RMNR}$.
- La deuxième partie est l'extraction des $\mathcal{RMNR}$ à partir du contexte formel et à l'aide des méthodes d'ACF (Figure 4.6, *(ii) Fouille*).
- Enfin, la partie finale de l'AAR est la transformation des règles en concepts, rôles, et assertions de rôles qui, s'ils sont jugés intéressants vis à vis des objectifs de l'expert et des connaissances présentes dans la BC de départ, y seront insérés (Figure 4.6, *(iii) Interprétation*).

L'"itération suivante de l'AAR pourra alors prendre en entrée l'ontologie ainsi raffinée.

Nous fixons un ensemble de constructeurs minimal obligatoire pour la LD utilisée puisque que la mise en œuvre de l'AAR nécessite les constructeurs de conjonction, existentiel, nominal et de rôle inverse, ce qui correspond à la LD $\mathcal{ELOI}$. La seule limite à l'utilisation de LD plus expressives est la complexité associée à la mise en œuvre des mécanismes de raisonnement utilisés (notamment la recherche d'instances).

Les sections suivantes détaillent les étapes permettant la mise en oeuvre de ces opérations, et pour chaque étape, le rôle de l'analyste.

2.3.2 L'AAR étape par étape

Etapes préliminaires : construction et peuplement d'une Base de Connaissances La construction d'ontologies et le peuplement de la BC associée à partir du contenu de bases de données ont été développés dans le chapitre 4.

Etape 1 : Sélection des instances de C_0

La première étape d'une itération d'AAR est la description en LD, par l'analyste, d'un concept C_0. Il n'y a pas de contrainte particulière concernant la définition de C_0 : ce peut être le concept $\top$, un

concept atomique ou défini dans la BC, ou encore la description d'un concept non nommé dans la BC mais défini pour l'occasion à partir de la BC et des constructeurs disponibles dans la logique choisie ($\mathcal{ELOI}$, $\mathcal{SHOIN}(\mathcal{D})$ par exemple).

La définition de C_0 sert à délimiter les assertions qui seront analysées et les concepts de la BC qui seront concernés par l'extraction de connaissances.

La description de C_0 permet d'obtenir un ensemble d'individus $\mathcal{A}_0$ instances de C_0 (voir définition 4.5). Ces individus constituent la base de l'analyse puisque l'approche va s'attacher à étudier comment ceux-ci sont reliés aux autres individus de la BC et à mettre en évidence des régularités remarquables dans l'ensemble de ces relations.

Etape 2 : Transformation des connaissances : exploration des *graphes d'assertions*

L'objectif de l'étape de transformation est de représenter dans un contexte formel (*i.e.* des données) les connaissances relatives aux relations de chaque individu de $\mathcal{A}_0$ avec l'ensemble des autres individus de la BC considérée. Pour ce faire nous définissons la notion de *graphe d'assertions*.

Définition 4.8 (Graphe d'assertions) *Soit* a *un individu de la* BC O. $G_a(V, E)$ *est un graphe d'origine* $a \in V$, *étiqueté, cyclique appelé le* graphe d'assertions *de* a *dans* O *où*

- V *est l'ensemble des nœuds de* G_a, *où chaque nœud* v *est un individu de* O *relié à* a *directement ou indirectement par un arc* E,
- E *est l'ensemble des arcs de* G_a, *où chaque arc* $E(v_1, v_2)$ *partant de* v_1 *vers* v_2 *est une assertion du rôle* E *entre les individus* v_1 *et* v_2 *dans* O. *Les arcs sont nommés par le nom du rôle dont ils représentent une instance. L'arc* $E(v_1, v_2)$ *peut être parcouru en sens inverse, de* v_2 *vers* v_1, *on note alors* $E^-(v_2, v_1)$. *Les arcs sont nommés par le nom du rôle dont ils représentent une instance.*

Le graphe G_a de a dans O contient l'ensemble des chemins possibles entre l'individu a et tout autre individu v de O auquel il est relié directement ou indirectement par n rôles E_i ($n \in \mathbb{N}$). De cette façon, chacune des relations existant entre a et les autres individus de O est représentée dans le graphe par un chemin de a vers un autre individu v, nœud de G_a.

Propriété 4.1 (Interprétation d'un chemin dans G_a) *S'il existe un chemin entre l'individu* a *et l'individu* v *passant successivement par les rôles* $E_1, E_2, ..., E_n$ *alors cela signifie que* a *est instance d'un concept noté* C_a *de forme* $\exists\, E_1 \circ E_2 \circ ... \circ E_n\{v\}$ *ou encore*

$$\exists\, E_1 \circ E_2 \circ ... \circ E_n\{v\}\,(a) \tag{4.6}$$

Propriété 4.2 *Soit* C *un concept,* R *un rôle et* a, v *deux individus de la* BC O. *Alors si*

$$O \models \exists\, R\,\{v\}\,(a) \quad et \quad O \models C(v)$$

alors $\exists\, R\,\{v\} \sqsubseteq \exists\, R.C$ *et donc*

$$O \models \exists\, R.C\,(a) \tag{4.7}$$

Nous proposons, pour chaque individu $a_i \in \mathcal{A}_0$ de parcourir, selon un algorithme simple, tous les chemins et sous-chemins possibles dans son graphe d'assertions G_{a_i}. L'objectif est d'associer à chaque individu a_i un ensemble de chemins, donc selon la Propriété 4.1 un ensemble de descriptions de concepts $C_{a_i,j}$ dont a_i est instance. A partir de cette association nous proposons de construire un contexte formel dont chaque objet fait référence à un individu $a_i \in \mathcal{A}_0$ et dont les attributs font référence aux différents concepts $C_{a_i,j}$ dont les a_i sont instances.

Pour explorer l'ensemble des chemins possibles dans les graphes d'assertions nous utilisons un algorithme de parcours en profondeur (décrit en Annexe F) fonction d'un paramètre, la *profondeur maximale* du parcours p_{max} définie par l'analyste en début de processus, et de deux restrictions :

(1) un même chemin ne peut pas passer deux fois par le même nœud,
(2) après avoir emprunté un arc qui correspond à un rôle E, l'algorithme interdit lors de l'étape suivante, d'emprunter un arc de même label en sens inverse, qui correspond au rôle inverse $\mathtt{E}^{-}$.

Le paramètre p_{max} limite le nombre maximum d'arcs qu'un seul chemin peut contenir et limite ainsi la progression en profondeur de l'algorithme. La première contrainte (1) garantie l'absence de cycle dans les chemins parcourus. La seconde contrainte (2) est un choix heuristique qui limite la taille finale du contexte formel généré.

Dans ce dernier cas et dans la limite de la profondeur maximale, il peut être démontré que l'algorithme parcourt de façon complète le graphe d'assertions *i.e.* parcourt tous les nœuds et arcs éloignés de moins de p_{max} arcs [RN03].

A la fin du parcours de graphes d'assertions des individus de $\mathcal{A}_0$, à chaque individu $\mathtt{a}_i \in \mathcal{A}_0$ est associé un ensemble de chemins et donc un ensemble de concepts $\mathtt{C}_{a_i,j}$ dont $\mathtt{a}_i$ est instance. A partir de cette association est alors construit un contexte formel $\mathcal{K}(\mathcal{G}, \mathcal{M}, \mathcal{I})$.

– Chaque individu $\mathtt{a}_i$ entraîne la création d'un *objet* $g_i \in \mathcal{G}$ dont le nom est celui de $\mathtt{a}_i$.

– Chaque concept $\mathtt{C}_{a_i,j}$ dont $\mathtt{a}_i$ est instance entraîne la création :
⇒ d'un *attribut* $m_{\mathtt{v}} \in \mathcal{M}$ dont le nom est la description en LD du concept $\mathtt{C}_{a_i,j}$.
A la notation classique
$$\exists\, E_1 \circ E_2 \circ ... \circ E_n \,\{\mathtt{v}\},$$
nous préférons ici la notation équivalente qui utilise le constructeur de concept nominal suivant (appelé *filler* en anglais)
$$E_1 \circ E_2 \circ ... \circ E_n : \mathtt{v}$$
plus court et plus simple à transformer en une chaîne de caractères. Ainsi, le nom de $m_{\mathtt{v}}$ est de la forme E1_o_E2_o_..._o_En : v. Lorsque $\mathtt{C}_{a_i,j} \equiv \mathtt{C}_{a_k,j}$, *i.e.* lorsque les individus $\mathtt{a}_i$ et $\mathtt{a}_k$ sont instances d'un même concept, alors $m_{\mathtt{v}}$ n'est créé que la première fois.

⇒ d'une *relation* $g_i \mathcal{I} m_{\mathtt{v}}$ entre l'objet g_i et l'attribut $m_{\mathtt{v}}$,

⇒ de n attributs $m_{\mathtt{C}}$ dont le nom est de la forme E1_o_E2_o_..._o_En : Cv par similarité avec le nom de l'attribut $m_{\mathtt{v}}$ mais où Cv fait référence à un concept $\mathtt{C}_{\mathtt{v}}$ dont v est instance,

⇒ des n relations $g_i \mathcal{I} m_{\mathtt{C}}$ correspondantes.

Les attributs $m_{\mathtt{C}}$ et les relations relatives $g_i \mathcal{I} m_{\mathtt{C}}$ sont créés dans le but d'augmenter le nombre d'attributs et de relations dans le contexte $\mathcal{K}$. Leur création suit la Propriété 4.2 qui dit que si v est instance d'un concept quelconque $\mathtt{C}_{\mathtt{v},j}$, alors a est également instance de $\exists E_1 \circ E_2 \circ ... \circ E_n.\mathtt{C}_{\mathtt{v}}$. Ainsi pour chaque concept $\mathtt{C}_{\mathtt{v},j}$ dont v est instance et tant que $\mathtt{C}_{\mathtt{v},j}$ ne présente pas de sous-concept dont v est également instance, *i.e.*

$$\not\models \mathtt{C}_{\mathtt{v},k} \sqsubseteq \mathtt{C}_{\mathtt{v},j},\ \forall k \neq j, \tag{4.8}$$

les attributs $m_{\mathtt{C}}$ et les relations $g_i \mathcal{I} m_{\mathtt{C}}$ correspondants sont ajoutés au contexte $\mathcal{K}$.

Le Tableau 4.3 représente le contexte formel $\mathcal{K}$ résultant de l'exploration des graphes d'assertions des individus de $\mathcal{A}_0$.

La taille du contexte généré selon l'exploration de graphes d'assertions dépend :

$\mathcal{G}$ \ $\mathcal{M}$	$m_{\mathsf{v}1}$	$m_{\mathsf{C}1,1}$	...	$m_{\mathsf{C}1,q_1}$	...	$m_{\mathsf{v}j}$	$m_{\mathsf{C}j,k}$	...	$m_{\mathsf{v}m}$	$m_{\mathsf{C}m,1}$	...	$m_{\mathsf{C}m,q_m}$
g_1	×	×		×		×	×		×	×		×
...												
g_i						×	×					
...												
g_n	×	×		×					×	×		×

TAB. 4.3 – *Contexte formel* $\mathcal{K}(\mathcal{G}, \mathcal{M}, \mathcal{I})$ résultat de l'exploration des graphes d'assertions

– suivant la dimension des objets, du nombre d'individus dans $\mathcal{A}_0$ (n dans le Tableau 4.3), et
– suivant la dimension des attributs, premièrement du nombre de nœuds dans le graphe lui même dépendant de la valeur de p_{max} (m dans le Tableau 4.3) ; et deuxièmement du nombre de concepts non ascendants (voir Équation 4.8) dont les individus v sont instances (q dans le Tableau 4.3).

Etape 3 : Analyse du contexte formel : ACF et $\mathcal{RMNR}$

Les méthodes d'Analyse de Concepts Formels (ACF), introduites dans la section 1.3.1 du chapitre 2, permettent la construction d'une représentation des données étudiées sous la forme d'un treillis de concepts, *i.e.* un ensemble de concepts formels organisés selon une structure hiérarchique. Cette structure résulte d'une analyse automatique des régularités existantes entre données : ces régularités résultent du fait que des objets distincts ont des attributs en commun et inversement, que des attributs distincts sont présentés par un même objet.

L'objectif de notre approche est justement de comparer la représentation résultant du processus automatique d'ACF à la représentation résultant d'un processus de modélisation dirigé par l'humain : l'ontologie de domaine. Pour ce faire nous proposons dans un premier temps de construire le treillis, puis d'utiliser les $\mathcal{RMNR}$ pour caractériser l'organisation en concepts formels du treillis afin, lors des étapes suivantes, de permettre la comparaison de cette représentation à celle de l'ontologie associée à la BC.

La construction du treillis peut être réalisée par l'utilisation de l'algorithme classique *Next Closure Algorithm* décrit par Ganter [Gan84]. Kuznetsov et Obiedkov ont récemment affiné cet algorithme et réalisé une comparaison des différentes méthodes de construction de treillis dans [KO02].

Une fois le treillis de concepts construit, son organisation peut être caractérisée selon différentes mesures ou méthodes. La mesure du *stabilité* d'un treillis, proposée par Kuznetsov [Kuz07] permet de caractériser la façon dont la description (le couple extension, intension) d'un concept est dépendante de chacun des objets qui compose son extension. Cette mesure a été utilisée par Jay *et al.* [JKN08] pour décrire des groupes sociaux d'intérêt à partir de concepts formels stables, *i.e.* dont l'existence ne repose pas uniquement sur quelques facteurs spécifiques. Nous proposons une méthode différente : l'utilisation des *Règles Minimales Non-Redondantes Réduites* ($\mathcal{RMNR}$ voir section 1.3.2) pour caractériser les aspects du treillis qui nous intéressent plus particulièrement, c'est à dire les relations entre les concepts formels et le nombre d'objets qui participe à la définition des concepts et des relations.

La recherche des Règles d'Associations (RA) est un moyen d'extraire d'un treillis ce genre d'information de façon exhaustive. Cependant, les RA produites présentent l'inconvénient d'être particulièrement volumineuses et redondantes. Nous préférons donc nous limiter à l'extraction d'une famille particulière de RA : les $\mathcal{RMNR}$. Ce type de règles présentent un premier avantage qui est d'être un sous-ensemble des RA réduit et concis, ce qui facilite l'étape suivante d'interprétation des règles par l'analyste. En effet, l'ensemble des $\mathcal{RMNR}$ est le plus petit ensemble de règles suffisant pour générer l'ensemble des RA.

Le deuxième avantage des $\mathcal{RMNR}$ est d'être particulièrement représentatives de la structure du treillis puisqu'une $\mathcal{RMNR}$ est produite à partir de la description d'un seul concept ou du regroupement de deux concepts directement reliés dans le treillis (*i.e.* un concept et son super- ou son sous-concept).

REMARQUE : En fonction de l'objectif de l'analyste, il peut être intéressant, non pas de chercher les règles fréquentes d'un contexte (RA ou $\mathcal{RMNR}$) mais de chercher un autre type de régularité, comme par exemple les règles rares [SNV07]. De plus, l'utilisation d'autres méthodes de fouille, comme le clustering hiérarchique [Fis87], peut proposer une représentation des données suivant une organisation différente de celle du treillis qu'il est également pertinent, selon les objectifs, de comparer à l'ontologie de domaine.

Etape 4 : Interprétation des régularités en terme de concepts et de rôles

A l'inverse de l'étape précédente, qui est automatique, cette étape implique l'analyste dès son début. L'analyste doit choisir, pour chaque $\mathcal{RMNR}$ et parmi les attributs $\mathcal{M}_r$ qui la composent, un ensemble d'attributs $\mathcal{M}_s \subseteq \mathcal{M}_r$ pertinents qui servira de base à la création de nouveaux concepts, de nouveaux rôles et/ou de nouvelles assertions de rôles.

Etape 4.a : Description de nouveaux concepts Selon un mécanisme inverse à celui opéré durant la phase de préparation (Figure 4.6, *(i)*), les attributs sélectionnés au sein d'une règle sont traduits en la description en LD du concept auquel ils font référence. Ainsi on distingue les attributs

$m_{\mathtt{v}}$ avec un nom de la forme $\mathtt{E1\_o\_E2\_o\_...\_o\_En : v}$ qui sont traduits en $\exists E_1 \circ E_2 \circ ... \circ E_n \{\mathtt{v}\}$,
$m_{\mathtt{C}}$ avec un nom de la forme $\mathtt{E1\_o\_E2\_o\_...\_o\_En : Cv}$ qui sont traduits en $\exists E_1 \circ E_2 \circ ... \circ E_n.\mathtt{Cv}$.

Un nouveau concept $\mathtt{C_{new}}$ est alors défini par la *conjonction* des descriptions de concepts correspondant aux attributs $\mathcal{M}_s$ d'une même règle. Par exemple, si dans la règle de la forme $<m_b \rightarrow m_d, m_e, m_f>$, les deux attributs m_b et m_d ont été sélectionnés (*i.e.* $\mathcal{M}_s = \{m_b, m_d\}$),

m_b nommé $\mathtt{R\_o\_S : b}$ est traduit en $\exists R \circ S \{\mathtt{b}\}$,
m_d nommé $\mathtt{T\_o\_U\_o\_V : d}$ est traduit en $\exists S \circ T \circ U \{\mathtt{d}\}$,

et leur conjonction permet de définir

$$\mathtt{C_{new}} \equiv \exists R \circ S \{\mathtt{b}\} \sqcap \exists S \circ T \circ U \{\mathtt{d}\} .$$

Etape 4.b :Création de nouveaux rôles et/ou d'assertions de rôles Les attributs sélectionnés par l'analyste $\mathcal{M}_s$ permettent également la création de nouveaux rôles et/ou assertions de rôles. Dans cet objectif, ces attributs sont traduits de la même façon que pour la création de nouveaux concepts, hormis le fait qu'ils ne sont pas associés pour créer une nouvelle description, et que seuls les concepts dont la description met en jeu le *constructeur nominal* (*i.e.* $\exists \mathtt{R}\{\mathtt{v}\}$ ou $\mathtt{R : v}$) sont utilisés. Si parmi les descriptions de concepts traduits depuis les attributs de $\mathcal{M}_s$, au moins deux font intervenir un constructeur nominal avec deux *nominaux* différents, alors chaque paire de *nominaux* est utilisée pour la construction d'un rôle et de deux assertions de rôles dans la BC. Ainsi pour chaque paire de nominaux, nous définissons $R_{\mathtt{new}}$ un rôle atomique dont le domaine et le co-domaine sont les concepts les plus spécifiques dont les nominaux sont instances, et deux assertions de rôle, la première de $R_{\mathtt{new}}$ et la seconde de son inverse $R^-_{\mathtt{new}}$ par le couple de nominaux.

Par exemple, si m_b et m_d sont deux attributs sélectionnés dans une même règle, alors la paire (b, d) qu'ils constituent est à la base de la définition du rôle atomique $R_{\mathtt{new}}$ dont le domaine et le co-domaine

sont respectivement le concept le plus spécifique dont b est instance et le concept le plus spécifique dont d est instance. La paire (b, d) est utilisée pour deux instanciations de rôles : $R_{\mathtt{new}}(\mathtt{b},\mathtt{d})$ et $R^{-}_{\mathtt{new}}(\mathtt{d},\mathtt{b})$.

REMARQUE : Les nominaux peuvent être instances de plusieurs concepts sans qu'il ne soit possible d'établir une relation de subsomption entre ces concepts (*i.e.* C(a) et D(a), mais $\not\models \mathtt{D} \sqsubseteq \mathtt{C}$ et $\not\models \mathtt{C} \sqsubseteq \mathtt{D}$). Dans ce cas, il n'existe pas un concept unique plus spécifique et l'analyste est sollicité pour statuer sur le concept à choisir entre C et D pour le domaine (ou le co-domaine) de $R_{\mathtt{new}}$.

Dans le cadre de notre approche, nous utilisons les règles ($\mathcal{RMNR}$) comme un moyen de caractérisation de la structure du treillis. La sémantique attachée à une règle est utilisée pour caractériser l'extension d'un concept formel (pour les règles certaines) et les relations avec ses concepts voisins (pour les règles approximatives). Cependant elle n'est pas utilisée directement pour définir des axiomes d'inclusion ($\sqsubseteq$) mais des axiomes assertionels (*i.e.* les assertions de rôles). En revanche l'étape suivante permet l'insertion des nouveaux concepts dans la BC initiale par la description d'axiomes d'inclusion.

Etape 5 : Insertion des nouvelles connaissances

Il s'agit dans cette étape de comparer les concepts et rôles ($\mathtt{C}_{\mathtt{new}}$ et $R_{\mathtt{new}}$) créés lors de l'étape précédente à ceux existants dans la BC de départ. Cette comparaison détermine si les nouveaux concepts et rôles n'existent pas déjà dans la BC (*i.e.* qu'ils sont véritablement nouveaux) et, dans le cas négatif, permet de définir la façon de les insérer de façon cohérente dans la BC.

Etape 5.a : Insertion de concepts Le *subsumant le plus spécifique* $\mathtt{C}_{\mathtt{subs}}$ du concept $\mathtt{C}_{\mathtt{new}}$ proposé est recherché dans l'ontologie associée à la BC. Si $\mathtt{C}_{\mathtt{new}} \equiv \mathtt{C}_{\mathtt{subs}}$, le concept existe déjà dans l'ontologie et $\mathtt{C}_{\mathtt{new}}$ n'est pas ajouté à l'ontologie. Sinon $\mathtt{C}_{\mathtt{new}} \sqsubseteq \mathtt{C}_{\mathtt{subs}}$ (sans que $\mathtt{C}_{\mathtt{subs}} \sqsubseteq \mathtt{C}_{\mathtt{new}}$) alors l'analyste a deux alternatives concernant la façon d'insérer le nouveau concept :

- selon l'analyste, $\mathtt{C}_{\mathtt{new}}$ est effectivement un sous-concept de $\mathtt{C}_{\mathtt{subs}}$. $\mathtt{C}_{\mathtt{new}}$ est inséré par l'ajout dans l'ontologie de l'axiome suivant : $\mathtt{C}_{\mathtt{new}} \sqsubseteq \mathtt{C}_{\mathtt{subs}}$. L'analyste peut alors attribuer un nom $\mathtt{C}_{\mathtt{new}}$.
- selon l'analyste, les définitions de l'ontologie de départ ne sont pas parfaites et $\mathtt{C}_{\mathtt{new}}$ est une description plus fine (ou plus exacte) de ce qui est censé être représenté par le concept $\mathtt{C}_{\mathtt{subs}}$. Dans ce cas $\mathtt{C}_{\mathtt{new}}$ est ajouté à l'ontologie par l'axiome suivant : $\mathtt{C}_{\mathtt{new}} \equiv \mathtt{C}_{\mathtt{subs}}$.

Etape 5.b : Insertion de rôle Selon l'existence ou non dans l'ontologie de rôles avec les mêmes domaine et co-domaine que $R_{\mathtt{new}}$, une suite d'opérations différentes est mise en œuvre . Dans le premier cas où de tels rôles existent déjà, l'analyste est sollicité. Si un des rôles de la liste correspond à la sémantique souhaitée pour *Rnew* il le choisit. Aucun rôle n'est créé dans l'ontologie, le rôle choisi et son inverse sont alors instanciés. En revanche, si aucun rôle de la liste n'est satisfaisant, un nouveau rôle est créé puis instancié.

Dans le second cas où aucun rôle existant ne partage les domaine et co-domaine de $R_{\mathtt{new}}$, un nouveau rôle est automatiquement créé et instancié. L'analyse n'intervient que pour nommer le nouveau rôle.

Enfin une classification d'instances par les mécanismes de raisonnement classiques sur la BC raffinée permet d'instancier les concepts $\mathtt{C}_{\mathtt{new}}$ avec les individus qui en sont instances.

Les deux dernières étapes *i.e.* l'interprétation des règles en termes de concepts et rôles en LD, puis leur insertion par la définition de nouveaux axiomes dans l'ontologie associée à la BC, sont formalisées dans deux algorithmes présentés ci après : le premier (Algorithme 4.1) décrit l'interprétation des règles en terme de nouveaux concepts de la BC et le second (Algorithme 4.2) décrit l'interprétation des règles

en de nouveaux rôles et assertions de rôles.

Algorithme 4.1 Depuis les attributs sélectionnés dans une règle à un nouveau concept

Entrée : $O = (\mathcal{T}, \mathcal{A}), \mathcal{K} = (\mathcal{G}, \mathcal{M}, \mathcal{I}), \mathcal{M}_0$ {ontologie, contexte, et attributs sélectionnées}
pour chaque m_i **de** $\mathcal{M}_0$
  si $\mathtt{C_{new}} \equiv \bot$ {nouveau concept en DL}
    $\mathtt{C_{new}} \coloneqq \mathtt{toDL}(m_i)$ {toDL retourne la déscription en DL}
  sinon
    $\mathtt{C_{new}} \coloneqq \mathtt{C_{new}} \sqcap \mathtt{toDL}(m_i)$
  fin si
fin pour chaque
si $\nexists\, \mathtt{D} \in \mathcal{T} : \mathtt{C_{new}} \equiv \mathtt{D}$ {si $\mathtt{C_{new}}$ n'exist pas ?}
  $\mathtt{C_{subs}} \coloneqq \mathtt{subs}(O, \mathtt{C_{new}})$ {subs retourne le subsumant direct de $\mathtt{C_{new}}$}
  Question à l'analyste
  selon analyste
    cas 1 {insert un nouveau concept}
      $\mathtt{C_{new}} \sqsubseteq \mathtt{C_{subs}}$
    cas 2 {complète la definition de concept}
      $\mathtt{C_{subs}} \equiv \mathtt{C_{subs}} \sqcap \mathtt{C_{new}}$
  fin selon
fin si
Sortie : O {Ontologie raffinée}

En bilan nous proposons la liste suivantes des étapes de l'AAR qui tirent parti des mécanismes de raisonnement associés à la BC :

- lors de la sélection des instances, la *recherche d'instances* permet de déterminer l'ensemble des individus instances du concept initial $\mathtt{C_0}$;
- lors de la transformation des connaissances, la *recherche du concept le plus spécifique* permet de déterminer l'identité du concept $\mathtt{C_v}$ dont v est instance pour la définition des attributs m_C de la forme `E1_o_E2_o_..._o_En` : Cv ;
- lors de l'insertion d'un nouveau concept dans la BC, la *classification des concepts* (et plus exactement la recherche du concept le plus spécifique) permet de positionner un nouveau concept défini $\mathtt{C_{new}}$ dans la hiérarchie de concepts ;
- après l'insertion de nouveaux concepts, de nouveaux rôles, et de nouvelles assertions de rôles, la *classification d'instances* permet de déterminer pour l'ensemble des individus s'ils sont instances d'un nouveau concept, et pour les individus impliqués dans une nouvelle assertion de rôle s'ils sont instances d'un concept ancien ou nouveau.

Les deux dernières utilisations des mécanismes de raisonnement permettent d'inférer de nouveaux axiomes terminologiques et assertionnels, *i.e.* de nouvelles unités de connaissances, qui viennent raffiner

Algorithme G.2 Depuis les attributs sélectionnés à de nouveaux rôles

Entrée : $\mathcal{O} = (\mathcal{T}, \mathcal{A}), \mathcal{K} = (\mathcal{G}, \mathcal{M}, \mathcal{I}), \mathcal{M}_0$ {ontologie, contexte, et attributs sélectionnées}
$C_{new}, \mathcal{R}_0 \coloneqq \emptyset$ {ensembles de concepts et de rôles}
$\mathtt{C_{new}} \coloneqq \bot$ {nouveau concept}
pour chaque m_i **de** $\mathcal{M}_0$
  $\mathtt{C_{new}} \coloneqq \mathtt{toDL}(m_i)$ {toDL retourne la description en DL}
  $C_{new} \coloneqq C_{new} \cup \{\mathtt{C_{new}}\}$
fin pour chaque
pour chaque $\mathtt{C_i}$ **de** C_{new}
  pour chaque $\mathtt{C_j}$ **de** C_{new} {itérations emboîtées pour comparé chaque concept à tous les autres}
    si $\not\models_{\mathcal{O}} \mathtt{C_i} \equiv \mathtt{C_j}$
      $b \coloneqq \mathtt{getFiller}(\mathtt{C_i})$
      $c \coloneqq \mathtt{getFiller}(\mathtt{C_j})$ {getFiller retourne le "nominal" d'une description de concept}
      $\mathcal{R}_0 \coloneqq \mathtt{domCodom}(\mathcal{O}, \mathtt{C_{subs}}(b), \mathtt{C_{subs}}(c))$ {domCodom retourne l'ensemble des rôles avec domain et codomain}
      si $\mathcal{R}_0 = \emptyset$ {description de nouveaux rôles}
        $\mathtt{domain}(\mathtt{R}_{new}), \mathtt{codomain}(\mathtt{R}^-_{new}) \coloneqq \mathtt{C_{subs}}(b)$
        $\mathtt{domain}(\mathtt{R}^-_{new}), \mathtt{codomain}(\mathtt{R}_{new}) \coloneqq \mathtt{C_{subs}}(c)$
        Question à l'analyste si $\mathtt{R}_{new}$ est pertinent
        si pertinent
          $\mathcal{T} \coloneqq \mathcal{T} \cup \{\mathtt{R}_{new}, \mathtt{R}^-_{new}\}$ {nouvau rôles}
          $\mathcal{A} \coloneqq \mathcal{A} \cup \{\mathtt{R}_{new}(b, c), \mathtt{R}^-_{new}(c, b)\}$
        fin si
      sinon {un rôle existe}
        pour chaque $\mathtt{R}_k$ **de** $\mathcal{R}_0$
          Question à l'analyste est ce que $\mathtt{R}_k$ est pertinent ?
          si pertinent
            $\mathcal{A} \coloneqq \mathcal{A} \cup \{\mathtt{R}_k(b, c), \mathtt{R}^-_k(c, b)\}$
      fin si, fin pour chaque, fin si
fin pour chaque, fin pour chaque, fin si
Sortie : $\mathcal{O}$ {Ontologie raffinée}

la BC initiale. C'est l'insertion de ces nouveaux axiomes dans la BC qui nous permet de dire que la méthode d'AAR autorise la découverte de connaissances implicites et nouvelles.

La description de la méthode d'AAR et son illustration avec un exemple pharmacogénomique ont donné lieu à deux publications [CSTND08b] et [CSTND08a].

La section 2.4 présente une expérimentation de la méthode d'AAR menée à partir de connaissances pharmacogénomiques.

2.4 Expérimentation en pharmacogénomique

Cette section présente une expérimentation de la méthode d'Extraction de Connaissance *à partir* d'une Base de Connaissances (ECBC) : l'AAR. Cette expérimentation commence par le peuplement d'une BC pharmacogénomique réalisé à l'aide de l'outil iSO-Pharm (présenté section 3.2, chapitre 3) puis continue par la mise en œuvre, *à partir* de cette BC, de la méthode d'AAR (présentée section 2.3 de ce chapitre).

Cette expérimentation s'intéresse aux données résultant d'une investigation clinique des réponses de patients asthmatiques à un médicament appelé le *montelukast*. Le déroulement de notre expérimentation est développé ci-dessous avec l'objectif d'illustrer et évaluer la capacité de l'AAR à *(1)* retrouver les résultats obtenus avec des méthodes statistiques classiques et *(2)* extraire de nouvelles connaissances. Notre motivation n'est pas de discuter les résultats de l'investigation initiale, mais plutôt de donner une deuxième vie à ces résultats en les étudiant selon une perspective différente.

2.4.1 Sources de données et de connaissances

Investigation clinique du montelukast La principale source de données exploitée correspond aux données recueillies au cours d'une investigation clinique menée dans le cadre de l'étude de la diversité de réponses des patients asthmatiques au montelukast. Des premiers résultats de cette investigation ont été publiés en 2006 par le groupe d'investigateurs Lima *et al.* [LZG+06]. Ces résultats ont été mis en évidence à partir de données génétiques et cliniques recueillies sur un sous-ensemble du panel recruté pour cette investigation et constitué de 61 patients. Les variables mesurées pour ces patients correspondent aux génotypes de 26 SNP et à l'enregistrement de deux signes cliniques principaux :

- la survenue, ou non, d'une crise d'asthme durant les 6 mois de traitement, noté "Exa" pour *exacerbation* en anglais et pouvant prendre les valeurs Yes, No ;
- le pourcentage de modification, après 6 mois de traitement, du Volume Expiratoire Maximum Seconde[66] (VEMS ou *FEV* en anglais) mesuré par rapport au Volume Expiratoire Maximum Seconde prédit à 6 mois. Cet attribut est noté "Per" pour *percent change in %predicted FEV1* est un pourcentage divisé par cent. Ses valeurs sont comprises dans l'intervalle [-0,16 ;1,16].

Les SNP génotypés sont localisés sur cinq gènes impliqués dans la voie des leukotriènes[67] : *ABCC1*, *ALOX5*, *CYSLTR1*, *LTA4H*, et *LTC4S* localisés respectivement sur les chromosomes 16, 10, X, 5, et 12.

Autres sources de données Pour peupler la BC nous extrayons, en plus des données de l'investigation, des données des bases de données PharmGKB, dbSNP, OMIM, Gene, et KEGG Pathway relatives notamment aux gènes impliqués dans la voie des leukotriènes, leurs structures, leurs variations génomiques, les réseaux métaboliques dans lesquels ils sont impliqués.

2.4.2 Préparation des données

Intégration des données génotypiques et phénotypiques Les données génétiques et cliniques concernant les patients de l'investigation sont disponibles publiquement dans deux fichiers distincts dans la base de données PharmGKB[68] (présentée chapitre 1, section 3.2). Pour des raisons de confidentialité, les patients sont identifiés dans chacun de ces deux fichiers par un identifiant distinct. Une première étape de préparation des données est la mise en correspondance des données contenues dans ces fichiers. Celle-ci est possible à l'aide d'une table de correspondance entre les identifiants des patients.

[66] Le VEMS correspond au volume expiré pendant la première seconde d'une expiration forcée.
[67] http://www.medscape.com/viewarticle/444395_5
[68] http://www.pharmgkb.org/do/serve?objId=PA142628130

Discrétisation des attributs Nous discrétisons les valeurs numériques de l'attribut "Per" en deux classes. Les valeurs de "Per" inférieures ou égales à 0,8 sont transformées en "⩽0,08" et les valeurs supérieures à 0,8 en "⩾0,09". Ces deux nouvelles valeurs de "Per" sont transformées par le système d'AAR en deux valeurs qui sont retrouvées dans les résultats : respectivement "Per__-inf-0.08_" et "Per__0.09-inf_".

Peuplement d'une Base de Connaissances L'outil iSO-Pharm, introduit chapitre 3, section 3.2 est utilisé pour peupler une BC pharmacogénomique notamment à partir des données de l'étude issues de PharmGKB. Les 61 patients de l'étude et les données cliniques (phénotypiques et génotypiques) qui leur sont associées servent notamment à la création de 61 assertions du concept "patient", de 127 assertions du concept "clinical_item" ou de ses descendants, et de nombreuses assertions du rôle "presents_clinical_item". Ce dernier rôle permet d'associer les instances des concepts "patient" et "clinical_item" conformément aux données de l'investigation clinique. Les données des autres bases (dbSNP, OMIM, Gene, et KEGG Pathway) permettent d'instancier des concepts et des rôles relatifs aux variations génomiques, aux gènes, aux médicaments, aux phénotypes, et à des réseaux métaboliques.

2.4.3 Plug-in Protégé pour l'AAR

La version 4 de l'éditeur de BC Protégé[69] donne la possibilité d'interfacer avec les fonctionnalités natives de Protégé, des outils externes ou *plug-in*. La méthode d'AAR détaillée en section 2.3 de ce chapitre est implémentée sous la forme d'un plug-in de Protégé. Une copie d'écran de l'interface graphique de l'onglet associé au plug-in est représentée en Figure 4.7. Le plug-in, comme son interface, est divisé en trois parties distinctes qui permettent de réaliser respectivement les étapes de préparation (au centre de l'interface), de fouille (en haut à droite), et d'interprétation (en bas à droite) de l'AAR.

- La partie dédiée à la préparation permet de décrire un concept C_0 et de sélectionner ses instances, de définir une profondeur maximale d_{max}, et sur cette base de construire un contexte formel. Une fois le contexte construit, cette partie permet également de retirer du contexte les attributs qui ne semblent pas pertinents pour la fouille.
- La partie dédiée à la fouille permet de lancer une recherche des $\mathcal{RMNR}$ selon un support et une confiance minimums min_supp et conf_min. Notre plug-in utilise la boite à outils CORON pour rechercher ces règles particulières [Sza06].
- La partie dédiée à l'interprétation permet la visualisation des règles, la sélection de règles puis la sélection d'attributs au sein des règles sélectionnées. Les attributs sélectionnés servent alors à construire et insérer dans la BC initiale de nouveaux concepts, de nouveaux rôles, et de nouvelles instances de rôles.

2.4.4 Résultats

L'expérimentation menée est réalisée suivant plusieurs itérations du processus d'AAR sur la BC peuplée. Les résultats obtenus lors d'une itération dépendent des résultats des itérations précédentes. Pour cette raison, nous les détaillons dans l'ordre de leur apparition.

Première itération La première itération de l'AAR est menée avec les paramètres suivants :

- $\mathsf{C}_0 \equiv \texttt{patient} \sqcap \texttt{is\_enrolled\_in} : \texttt{montelukast\_study}$
- d_{max}= 2
- min_supp = 0,8

[69] `http://protegewiki.stanford.edu/index.php/Protege4UserDocs`

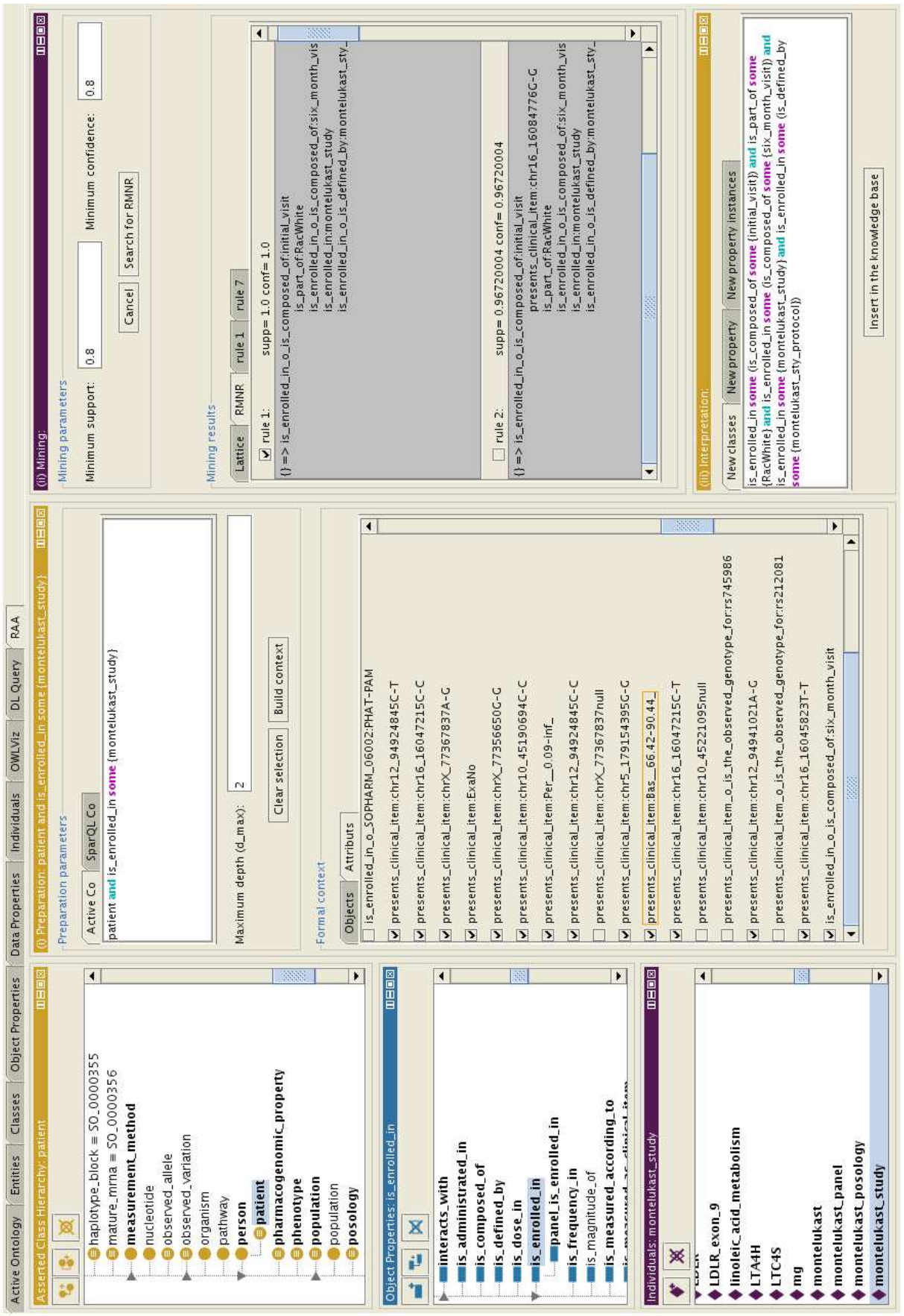

FIG. 4.7 – Capture d'écran du plugin de Protégé 4 pour l'Analyse d'Assertions de Rôles

– min_conf = 0,8

La première $\mathcal{RMNR}$ produite présente un support et une confiance de 1. Sa composition est la suivante :

Règle 1

{} =>	*{is_enrolled_in_o_is_composed_of :initial_visit,* *is_part_of :RacWithe,* *is_enrolled_in_o_is_composed_of :six_month_visit,* *is_enrolled_in :montelukast_study* *is_enrolled_in_o_is_defined_by :montelukast_sty_protocol}*

Le symbole { } (qui constitue la prémisse de la règle) représente l'ensemble de tous les attributs du contexte formel. Cette première règle, du fait que la confiance est égale à 1, peut être interprétée comme le fait que tous les individus instances de $\mathtt{C}_0$ sont aussi instances des concepts décrits par les attributs de la conclusion de la règle. Dans ce premier cas, tous les attributs nous intéressent pour constituer un nouveau concept. Alors, aucun attribut de la règle n'est exclu par l'utilisateur et la règle 1 est transformée par le système en LD, sous la forme de la définition de concept suivante :

$$\begin{aligned}\mathtt{C}_{\mathtt{new1}} \equiv\ & \mathtt{is\_enrolled\_in} \circ \mathtt{is\_composed\_of} : \mathtt{initial\_visit} \sqcap \\ & \mathtt{is\_part\_of} : \mathtt{RacWithe} \sqcap \\ & \mathtt{is\_enrolled\_in} \circ \mathtt{is\_composed\_of} : \mathtt{six\_month\_visit} \sqcap \\ & \mathtt{is\_enrolled\_in} : \mathtt{montelukast\_study} \sqcap \\ & \mathtt{is\_enrolled\_in} \circ \mathtt{is\_defined\_by} : \mathtt{montelukast\_sty\_protocol}\end{aligned}$$

On peut tout d'abord remarquer que la quatrième ligne de la définition de $\mathtt{C}_{\mathtt{new1}}$ correspond à une partie de la description de $\mathtt{C}_0$. De façon informelle, le concept $\mathtt{C}_{\mathtt{new1}}$ peut être interprété comme "l'ensemble des individus qui sont recrutés dans l'étude du montelukast, qui sont recrutés dans quelque chose qui est composé d'une visite initiale et d'une visite à six mois, qui sont d'une ethnie blanche[70], et qui sont recrutés dans quelque chose qui est défini par le protocole de l'étude du montelukast". Ceci correspond finalement à une description précise des patients qui sont impliquées dans l'étude du montelukast. Une telle description n'existe pas dans la BC dans laquelle la description des patients se limite à la définition du concept `patient` et à son concept parent `person`.

Alors, le nouveau concept $\mathtt{C}_{\mathtt{new1}}$ est inséré dans la BC. Pour cela, un nom plus explicite que $\mathtt{C}_{\mathtt{new1}}$ lui est attribué par l'utilisateur : `montelukast_study_patient`. Le système le branche dans un premier temps à la racine des concepts de la BC : $\top$. Dans un deuxième temps, l'utilisation du mécanisme de classification permet de proposer un nouveau positionnement au concept `montelukast_study_patient` dans la hiérarchie de concepts. Le résultat est le suivant :

$$\mathtt{montelukast\_study\_patient} \sqsubseteq \mathtt{patient}$$

Ce positionnement s'explique par *(1)* la définition du concept `patient` initiale dans l'ontologie SO-Pharm qui contient l'axiome

$$\mathtt{patient} \equiv \exists\, \mathtt{is\_enrolled\_in.clinical\_trial} \sqcup \exists\, \mathtt{is\_part\_of.clinical\_trial\_panel}\,,$$

et *(2)* l'axiome d'assertion

$$\mathtt{clinical\_trial(montelukast\_study)}$$

[70] La notion d'ethnicité est rapportée dans l'étude selon les recommandations de l'Institut National de la Santé états-unien (le NIH) : `http://grants2.nih.gov/grants/guide/notice-files/NOT-OD-01-053.html`.

selon lequel l'individu montelukast_study est une instance du concept clinical_trial suite au peuplement de la BC. A partir de ces deux éléments, le système peut détérminer que la définition du nouveau concept contient un condition suffisante pour inférer que toutes ses instances sont également instances de patient.

A ce niveau, l'utilisateur doit déterminer si le nouveau concept est une meilleure définition de ce qui devrait être représenté par son subsumant le plus spécifique ou bien si le nouveau concept est effectivement un sous-concept de celui-ci. De par le fait que le nouveau concept (montelukast_study_patient) est effectivement un concept différent de son subsumant le plus spécifique (patient), le nouveau concept est positionné dans la BC par l'insertion de l'axiome d'inclusion de concept proposé par le système.

Concernant la potentielle création de nouveaux rôles et de nouvelles assertions de rôles, les couples possibles entre les individus impliqués dans la règle sont examinés par le système. Il existe déjà des assertions de rôles entre initial_visit, six_month_visit, et montelukast_sty_protocol dans la BC, aussi la possibilité de créer des assertions associant ces individus est rejetée. Les relations possibles entre les autres individus ne paraissent pas suffisamment intéressantes à l'utilisateur pour donner lieu à la création de rôles ou d'assertions de rôles. Au final, aucun rôle ou assertion de rôle n'est créé à partir de cette règle.

Les paramètres de cette itération et son premier résultat sont illustrés sur la représentation de l'interface graphique du plug-in de Protégé dédié à l'AAR présenté dans la Figure 4.7.

Lors de cette première itération, la profondeur d_{max} utilisée est égale à 2. Aussi, les rôles impliqués dans la définition du concept sont constitués au maximum d'une seule composition de rôles. Augmenter la profondeur de recherche dans les graphes d'assertions permet de générer des attributs qui correspondent à des compositions multiples de rôles (par exemple 3 compositions de rôle au maximum pour d_{max}=4). Ceci est illustré par l'itération suivante du processus d'AAR qui aboutit à l'enrichissement de la définition de notre nouveau concept montelukast_study_patient.

Deuxième itération Les paramètres définis pour cette deuxième itération sont identiques à ceux utilisés précédemment, excepté pour la profondeur *dmax* qui est augmentée. De cette façon nous avons :

- $C_0 \equiv$ patient $\sqcap$ is_enrolled_in : montelukast_study
- d_{max}= 3
- min_supp = 0,8
- min_conf = 0,8

Préalablement à la fouille nous excluons un sous-ensemble d'attributs non pertinents afin de décharger le nombre d'attributs dans les règles. Une des premières règles générées, présentant un support et une confiance de 1, est la suivante :

Règle 2

{} => *{presents_clinical_item_o_is_the_observed_genotype_for_o_isVariantIn :CYSLTR1,*
presents_clinical_item_o_is_the_observed_genotype_for_o_isVariantIn :ALOX5,
presents_clinical_item_o_is_the_observed_genotype_for_o_isStoredInVarDb :ncbi_dbsnp_125,
presents_clinical_item_o_is_the_observed_genotype_for_o_isVariantIn :LTA4H,
presents_clinical_item_o_is_the_observed_genotype_for_o_isVariantIn :ABCC1,
is_enrolled_in :montelukast_study
presents_clinical_item_o_is_the_observed_genotype_for_o_isVariantIn :LTC4S,
is_enrolled_in_o_is_defined_by_o_is_composed_of :montelukast_treatment}

La sélection d'attributs explique notamment que les attributs de la règle 1 (exclus lors de cette nouvelle itération) n'apparaissent pas dans la règle 2 (sauf le sixième). En revanche les attributs ici associés

n'apparaissaient pas dans la règle 1 car les rôles qu'ils invoquent impliquent l'enchaînement de deux compositions de rôle.

Cette règle illustre le fait que chaque individu instance de C_0 est associé à des items cliniques qui sont des génotypes observés pour des variants, localisés sur cinq gènes *CYSLTR1, ALOX5, LTA4H, ABCC1, LTC4S*. Dans le cas de notre étude sur le montelukast, savoir que tous les patients ont des variants génotypés sur chacun de ces cinq gènes n'est pas une connaissance nouvelle puisque celle-ci est décrite dans les méta-données dont nous disposons à propos de l'étude (l'article de Lima *et al.* et les méta-données associées aux fichiers dans PharmGKB). Cependant, la régularité exprimée par cette règle peut permettre de formaliser cette connaissance de façon explicite dans la BC. Pour cette raison nous proposons un nouveau concept $\mathsf{C}_{\mathtt{new2}}$ sur la base de cette règle. Par ailleurs, il est facile d'imaginer des cas où les méta-données disponibles sur les variants explorés sont partielles ou inexistantes, ou encore des cas où le nombre de variants explorés est trop grand pour que ces méta-données soient facilement exploitables. Dans ces cas, le concept traduit à partir de cette règle peut être porteur d'une connaissance nouvelle.

$$
\begin{aligned}
\mathsf{C}_{\mathtt{new2}} \equiv\ & \mathtt{presents\_clinical\_item} \circ \mathtt{is\_the\_observed\_genotype\_for} \circ \mathtt{isVariantIn} : \mathtt{CYSLTR1} \sqcap \\
& \mathtt{presents\_clinical\_item} \circ \mathtt{is\_the\_observed\_genotype\_for} \circ \mathtt{isVariantIn} : \mathtt{ALOX5} \sqcap \\
& \mathtt{presents\_clinical\_item} \circ \mathtt{is\_the\_observed\_genotype\_for} \circ \mathtt{isVariantIn} : \mathtt{LTA4H} \sqcap \\
& \mathtt{presents\_clinical\_item} \circ \mathtt{is\_the\_observed\_genotype\_for} \circ \mathtt{isVariantIn} : \mathtt{ABCC1} \sqcap \\
& \mathtt{is\_enrolled\_in} : \mathtt{montelukast\_study} \sqcap \\
& \mathtt{presents\_clinical\_item} \circ \mathtt{is\_the\_observed\_genotype\_for} \circ \mathtt{isVariantIn} : \mathtt{LTC4S} \sqcap \\
& \mathtt{is\_enrolled\_in} \circ \mathtt{is\_defined\_by} \circ \mathtt{is\_composed\_of} : \mathtt{montelukast\_treatment}
\end{aligned}
$$

Nous remarquons que le troisième attribut impliqué dans la conséquence de la règle 2 n'a pas été utilisé par le système dans la définition du nouveau concept. Ceci est la conséquence de son exclusion par l'utilisateur comme le permet l'interface graphique du plug-in lors de l'interprétation des règles. La raison de ce choix dépend du contexte de l'expérimentation, pour lequel nous ne considérons pas pertinent le fait que tous les patients puissent être génotypés pour au moins un variant répertorié dans dbSNP.

Le positionnement proposé par le système pour ce nouveau concept dans la hiérarchie de concepts de la BC est

$$\mathsf{C}_{\mathtt{new2}} \sqsubseteq \mathtt{montelukast\_study\_patient}.$$

Dans ce cas, le nouveau concept ne constitue pas aux yeux de l'utilisateur un sous-ensemble des individus définis par le concept `montelukast_study_patient`, mais plutôt une description différente de ce concept. De fait, nous choisissons d'enrichir la définition de `montelukast_study_patient` comme le permet le système en positionnant $\mathsf{C}_{\mathtt{new2}}$ par l'ajout dans la BC de l'axiome suivant :

$$\mathsf{C}_{\mathtt{new2}} \equiv \mathtt{montelukast\_study\_patient}.$$

Aucun rôle n'est créé ou instancié à partir de cette règle.

Les deux premières itérations permettent d'illustrer deux fonctionnalités de l'AAR :

- premièrement, l'augmentation du paramètre d_{max} permet d'extraire des connaissances qui mettent en jeu des individus indirectement associés dans la BC. L'exemple présenté ici illustre la définition d'un nouveau concept à partir de connaissances construites avec des données de l'investigation du montelukast et d'annotations sur la position des variants issues de dbSNP.
- Deuxièmement, une nouvelle description de concepts peut venir soit s'insérer dans la hiérarchie de concepts ($\mathsf{C}_{\mathtt{new1}}$) soit enrichir un définition existante ($\mathsf{C}_{\mathtt{new2}}$).

Troisième itération Nous poursuivons l'exploration de la BC avec le même concept initial C_0, la même profondeur $d_{max} = 3$, mais nous diminuons le support minimum à 0,3. Les paramètres utilisés cette fois sont :

- $C_0 \equiv$ `patient` $\sqcap$ `is_enrolled_in` : `montelukast_study`
- d_{max}= 3
- min_supp = 0,3
- min_conf = 0,8

La recherche des $\mathcal{RMNR}$ révèle alors de nombreuses associations entre génotypes. Nous sélectionnons les règles qui associent des génotypes observés sur le même gène. La règle 3 (support=0,31, confiance=0,95) en est un exemple.

Règle 3

{presents_clinical_item :chrX_77389891A-G,
presents_clinical_item :chrX_77367837A-G} => *{presents_clinical_item :chrX_77334462A-G}*

Ce genre de règle nous intéresse particulièrement pour étudier les génotypes qui ségrègent ensemble, *i.e.* qui sont transmis de façon groupée à la manière des haplotypes. Dans leur travaux, Lima *et al.* mettent en évidence trois groupes de génotypes fortement associés par déséquilibre de liaison (*Linkage Desquilibrium* ou LD en anglais). Ceux-ci sont reportés dans la colonne de gauche du Tableau 4.4. Suivant notre méthode, nous isolons, parmi la centaine de règles produites, 7 règles qui nous permettent d'isoler 7 groupes. La règle 3 ci-dessus en est un exemple. L'ensemble de ces règles est reporté en Annexe H avec leurs supports et confiances. La colonne de droite du Tableau 4.4 représente les 7 groupes de génotype associés à partir de ces règles.

Par ailleurs, ce sont les associations entre les individus évoqués dans ces règles que nous souhaitons insérer dans la BC. Aussi pour chaque règle nous ne construisons pas un nouveau concept, mais cherchons à instancier des rôles entre les individus correspondant aux génotypes.

SO-Pharm ne dispose d'aucun rôle dont le domaine et le co-domaine sont définis par le même concept `genomic_genotype`, ce qui permettrait d'associer deux instances de ce concept auxquelles font référence les attributs des règles. Aussi notre système d'AAR propose automatiquement, lors du traitement de la première règle associant des génotypes, la création d'un nouveau rôle avec `genomic_genotype` en domaine et co-domaine. Celui-ci est appelé par l'utilisateur `segregates_with`. Ensuite, ce rôle et son inverse (*i.e.* lui même dans le cas particulier de `segregates_with`) sont instanciés avec chaque paire de génotypes inclus dans une règle. De cette façon la règle 3 entraîne, après validation de l'utilisateur, l'insertion dans la BC des 6 assertions de rôles suivantes :

`segregates_with` (`chrX_77389891A-G`, `chrX_77367837A-G`),
`segregates_with`$^-$(`chrX_77389891A-G`, `chrX_77367837A-G`),

`segregates_with` (`chrX_77389891A-G`, `chrX_77334462A-G`),
`segregates_with`$^-$(`chrX_77389891A-G`, `chrX_77334462A-G`),

`segregates_with` (`chrX_77367837A-G`, `chrX_77334462A-G`),
`segregates_with`$^-$(`chrX_77367837A-G`,, `chrX_77334462A-G`)

Au total, les 7 règles entraînent l'insertion dans la BC d'un nouveau rôle (`segregates_with`) et de 26 assertions de ce nouveau rôle.

Nous retrouvons les mêmes groupes que Lima *et Al.* Nos groupes sont plus restreints notamment pour le premier groupe du gène *ABCC1* et celui du gène *ALOX5*. Cependant nous mettons en évidence

Gène	Génotypes associés Lima *et al.* (LD)	Génotypes associées Analyse des Assertions de Rôles	
ABCC1	Chr16 :15986618G-G	Chr16 :15986618G-G	
	Chr16 :15994334C-C	Chr16 :15994334C-C	
	Chr16 :16016395A-A		
	Chr16 :16045823T-T	Chr16 :16045823T-T	Chr16 :16045823C-T
	Chr16 :16047215T-T	Chr16 :16047215T-T	Chr16 :16047215C-T
ALOX5	Chr10 :45190694C-T	Chr10 :45190694C-T	
	Chr10 :45211490A-G	Chr10 :45211490A-G	Chr10 :45211490A-G
	Chr10 :45221095A-A	Chr10 :45221095A-G	Chr10 :45221095A-G
	Chr10 :45198914A-G		Chr10 :45198914A-A
	Chr10 :45237098A-G		
CYSLTR1	∅	ChrX :77346486T-T	
		ChrX :77356650G-G	
		ChrX :77389891G-G	ChrX :77389891A-G
		ChrX :77367837A-A	ChrX :77367837A-G
		ChrX :77334462A-A	ChrX :77334462A-G
LTA4H	∅	∅	
LTC4S	∅	∅	

TAB. 4.4 – Groupes de génotypes associés au sein des gènes étudiés dans l'investigation clinique de Lima *et al.* [LZG+06]. La colonne de gauche présente les trois groupes de génotypes mis en évidence par Lima *et al.* par la mesure des déséquilibres de liaison (*Linkage Desquilibrium* ou LD en anglais). La colonne de droite présente les groupes que nous avons mis en évidence à partir du même jeu de données avec l'AAR . Cette deuxième colonne présente dans certains cas deux associations de génotypes différents pour un même groupe de variations (gène *ABCC1* et *CYSLTR1*). Les règles dont sont extraits ces 7 groupes sont reportées en Annexe H.

deux groupes particuliers qui correspondent à des allèles différents de variations déjà impliquées dans un groupe : le couple Chr16 :16045823C-T, Chr16 :16047215C-T et le triplet ChrX :77334462A-G, ChrX :77367837A-G, ChrX :77389891A-G. De plus, nous mettons en évidence une association entre deux groupes de génotypes du gène *CYSLTR1* absents des résultats de Lima *et al.* . Les supports et confiances de chaque règle, reportés en Annexe H, permettent de juger la fréquence dans la population de ces associations.

Quatrième itération Nous poursuivons encore l'exploration de la BC avec le même concept initial C_0 et la même profondeur d_{max} en diminuant le support minimum, cette fois, à 0,2. Les paramètres utilisés sont ainsi :

- $C_0 \equiv$ `patient` $\sqcap$ `is_enrolled_in` : `montelukast_study`
- d_{max}= 3
- min_supp = 0,2
- min_conf = 0,8

L'objectif de cette nouvelle itération est d'isoler des règles associant un attribut relatif au phénotype (Per= {"⩾0,09", "⩽0,08"} ou Exa={"No", "Yes"}) et un ou plusieurs attributs décrivant un génotype spécifique à ce phénotype.

Avec un support minimum de 0,2, le nombre de règles devient important (proche de 3000 règles) cependant les règles impliquant un attribut relatif au phénotype sont relativement rares dans cet ensemble (<5%). Pour trouver ces règles plus facilement nous utilisons un système de filtres simples, semblables à ceux décrits dans la section 1.4 du chapitre 2. Nous isolons au final 5 règles qui correspondent au modèle imposé par les filtres. La règle 4 (support=0,26, confiance=0,80) ci-dessous en est un exemple. L'ensemble des règles isolées est reporté en Annexe H.

Règle 4

{presents_clinical_item :chrX_77334462A-G} =>	*{presents_clinical_item :chrX_77367837A-G, presents_clinical_item :Per__-inf-0.08_, is_enrolled_in_o_is_defined_by_o_is_composed_of : montelukast_treatment}*

Le fait que la règle 2 ait un support et une confiance égaux à 1 permet de déduire que l'attribut de sa conclusion *is_enrolled_in_o_is_defined_by_o_is_composed_of : montelukast_treatment* est présent dans toutes les règles. Cet attribut n'apparaît pas dans la règle 3 car nous l'avions exclu de la recherche de règles comme le système le permet afin d'alléger le nombre d'attributs dans les règles. Nous nous permettons cette exclusion car d'une part nous savons que cet attribut est présent pour chaque objet du contexte et d'autre part lors de l'itération précédente, nous ne cherchions pas de règles impliquant un traitement mais uniquement des génotypes.

Des cinq règles isolées, nous sommes capables d'identifier quatre génotypes et une paire de génotypes spécifiques de trois phénotypes différents. La colonne de droite du Tableau 4.5 présente ces génotypes. La colonne de gauche de ce Tableau présente les résultats rapportés dans Lima *et al.* en utilisant deux tests statistiques : χ^2 et le rapport de vraisemblance. Avec l'AAR nous retrouvons deux des cinq génotypes qu'ils associent à un phénotype particulier (Chr5 :179153244A-C et Chr12 :94941021A-G). Les trois génotypes non retrouvés (Chr10 :45221095G-G, Chr16 :15994335C-T, et Chr12 :94941021G-G) sont observés dans Lima *et al.* sur des sous-groupes de patients particulièrement restreints (respectivement n = 6, 8, et 5). Les données manquantes et le seuil utilisé (0,08) pour discrétiser l'attribut "Per" ramènent dans notre jeu de données ces populations à n= 4, 5, et 5. Sur une population totale de 61 patients, la probabilité d'observation de ces génotypes en même temps que le phénotype associé est alors inférieure à 0,05. Pour les retrouver ensemble dans des règles, il nous faudrait réduire le support minimum en dessous de cette valeur . Ceci aurait pour conséquence un accroissement important du nombre de règles et par conséquent, du temps nécessaire pour isoler les règles pertinentes qui correspondent au modèle recherché.

Nous identifions cependant lors de cette itération quatre génotypes spécifiques à des phénotypes qui ne l'avaient pas été par Lima *et al.* .

Au niveau de la BC, chaque règle isolée permet la création d'un nouveau concept. L'utilisateur lui associe un nom, et le système l'insère dans la hiérarchie de concepts de la BC. Par exemple, la règle 4 permet de définir le concept suivant nommé `patient_with_low_chge_in_fev_grp2` par l'utilisateur :

$$\begin{aligned} \texttt{patient\_with\_low\_chge\_in\_fev\_2} \equiv\ & \texttt{presents\_clinical\_item} : \texttt{chrX\_77334462A-G} \sqcap \\ & \texttt{presents\_clinical\_item} : \texttt{chrX\_77367837A-G} \sqcap \\ & \texttt{presents\_clinical\_item} : \texttt{Per\_\_-inf-0.08\_} \sqcap \\ & \texttt{is\_enrolled\_in} \circ \texttt{is\_defined\_by} \circ \texttt{is\_composed\_of} : \texttt{montelukast\_treatment} \end{aligned}$$

De plus, pour chaque règle des assertions des sous-rôles du rôle `interacts_with` sont créées en fonction des concepts dont sont instances les paires d'individus considérés. De cette façon, la règle 4 permet d'insérer dans la BC, entre autres, les axiomes assertionnels suivants

Phénotype	Génotypes spécifiques Lima *et al.* (χ^2, rapport de vraisemblance)	Génotypes spécifiques Analyse des Assertions de Rôles
Per="⩾0,09"	Chr10 :45221095G-G Chr16 :15994335C-T	∅
Per="⩽0,08"	∅	Chr10 :45211490A-A ChrX :77334462A-G ChrX :77367837A-G
Exa="No"	Chr5 :179153244A-C	Chr5 :179153244A-C Chr16 :161443440C-G
Exa="Yes"	Chr12 :94941021A-G Chr12 :94941021G-G	Chr12 :94941021A-G

TAB. 4.5 – Génotypes spécifiques aux phénotypes présentés dans la colonne de gauche. La colonne du centre représente les génotypes spécifiques mis en évidence dans Lima *et al.* par méthodes statistiques(χ^2 et rapport de vraisemblance) [LZG+06]. La colonne de droite représente les variations mises en évidence par notre approche d'Analyse des Assertions de Rôles. Les règles qui mettent en évidence ces associations sont reportées en Annexe H.

`interacts_with_phenotype` (`chrX_77334462A-G`, `Per__-inf-0.08_`),
`interacts_with_phenotype`$^-$(`chrX_77334462A-G`, `Per__-inf-0.08_`),

`interacts_with_drug_treatment` (`chrX_77334462A-G`, `montelukast_treatment`),
`interacts_with_drug_treatment`$^-$(`chrX_77334462A-G`, `montelukast_treatment`),

ainsi que des axiomes de la même forme impliquant le second génotype (`chrX_77367837A-G`), et d'autres à partir de `Per__-inf-0.08_` et de `montelukast_treatment`.

Au final, les cinq règles sont à l'origine de cinq définitions de concepts et 68 assertions de rôles insérés dans la BC.

La classification d'instances sur la BC permet de représenter explicitement les génotypes, phénotypes, et traitements qui interviennent dans une réaction pharmacogénomique à un traitement. SO-Pharm contient initialement trois concepts définis de façon symétrique

- `pharmacogenomic_genotype_item`,
- `pharmacogenomic_phenotype_item`, et
- `pharmacogenomic_drug_treatment`.

Par exemple un génotype qui intervient dans une réaction pharmacogénomique est défini comme un génotype qui interagit à la fois avec un phénotype et un traitement de la façon suivante :

$$\texttt{pharmacogenomic\_genotype\_item} \equiv \geqslant 1\ \texttt{interacts\_with\_phenotype} \sqcap \geqslant 1\ \texttt{interacts\_with\_drug\_treatment}$$

Ainsi, à partir de cette définition il est inféré que l'individu `chrX_77334462A-G` déjà instance de `genotype_item` est également instance de `pharmacogenomic_genotype_item`.

De la même façon, la classification qui termine cette itération permet de statuer sur le fait que

- 6 individus instances de `genotype_item` sont aussi instances de `pharmacogenomic_genotype_item`,

– 4 individus instances de `phenotype_item` sont instances de `pharmacogenomic_phenotype_item`,
– 1 individu instance de `drug_treatment` (`montelukast_treatment`) est également instance de `pharmacogenomic_drug_treatment`.

2.4.5 Bilan et discussion

L'AAR nous a permis d'extraire et de formaliser un certain nombre d'*unités de connaissances* soit sous la forme d'axiomes terminologiques (*i.e.* impliquant $\equiv$, ou $\sqsubseteq$), soit sous la forme d'axiomes assertionnels (*i.e.* les assertions de rôles et d'instances). Ces unités de connaissances n'étaient préalablement pas présentes dans la BC par conséquent nous les qualifions de *nouvelles*. En outre, l'analyste a lui même jugé intéressant de les insérer dans la BC, par conséquent nous les qualifions également de *pertinentes*. De fait, l'expérimentation montre que l'AAR permet de retrouver l'essentiel des résultats qui avaient été manuellement extrait par des méthodes statistiques classiques dans [LZG+06] : des associations fortes entre génotypes, des associations génotype–phénotype. Notre méthode permet d'aller plus loin dans l'exploitation des données analysées en isolant en plus de ces résultats : de nouvelles associations entre génotypes, de nouvelles relations génotype–phénotype, et des relations génotype–traitement–phénotype. L'ensemble de ces résultats est représenté de façon formelle dans la BC qui peut être enrichie avec de nouvelles données ou donner lieu à de nouvelles expérimentations.

L'expérimentation montre que la préparation des données, le paramétrage, l'exclusion d'attributs permettent d'orienter et de contrôler l'AAR. L'influence de ces différentes opérations sur les résultats de l'analyse est discutée dans la suite de cette section.

La discrétisation des valeurs de l'attribut "Per" effectuée lors de l'étape de préparation des données est un premier facteur jouant sur les résultats. En effet, le choix d'un seuil moins élevé pour la discrétisation, par exemple 0,04 au lieu de 0,08, permettrait d'augmenter le nombre d'objets qui présentent une valeur au dessus de ce seuil et par conséquent d'augmenter le nombre d'objets qui peuvent présenter à la fois une valeur de "Per" au dessus du seuil et un génotype particulier. La valeur choisie pour ce seuil explique en partie pourquoi, contrairement à Lima *et al.* , nous ne retrouvons pas de génotype spécifique au phénotype Per="$\geqslant$0,09". L'autre explication réside dans le faible nombre de cas sur lesquels se basent Lima *et al.* pour estimer ces associations.

Il apparaît au cours de l'expérimentation que l'exclusion des attributs les moins pertinents du contexte facilite l'étape d'interprétation. Le moyen proposé d'exclure des attributs dans l'implémentation actuelle est entièrement manuel. Il serait certainement intéressant d'adapter l'approche de sélection de données guidée par les connaissances, proposée dans la section 1 de ce chapitre, pour faciliter l'exclusion d'attributs du contexte manipulé en AAR.

Le nombre de règles produites est un facteur important de la difficulté à interpréter les résultats. Ce nombre de règle est tout d'abord sensible au nombre d'attributs considérés pour la recherche des règles, mais aussi sensible à d'autres paramètres. Ainsi, la profondeur d_{max} entraîne la constitution d'un contexte plus volumineux, et par conséquent, une production de règles souvent plus nombreuses. Enfin les support et confiance minimums permettent de moduler le nombre de règles. Hypothétiquement il pourrait également être envisagé de contraindre le parcours des graphes d'assertions de sorte à ce que seuls les chemins associés à une sémantique définie soient parcourus. En conséquence le contexte résultant ne présenterait que les attributs générés à partir du parcours de ces chemins spécifiques.

Voici un ordre de grandeur du nombre de règles produites lors des différentes itérations présentées :

première itération :	$<$ 10 règles
deuxième itération :	$<$ 20 règles
troisième itération :	$<$ 100 règles
quatrième itération :	$<$ 3000 règles.

Lorsque le support est diminué en deça de 0,2, le nombre de règles augmente davantage et l'interprétation devient délicate malgré l'utilisation de filtres. Ceci est en partie dû à la méthode de fouille utilisée dont l'objectif est la recherche de règles fréquentes. Dans le cas où les règles recherchées apparaissent avec un support de 0,1 l'utilisation d'une méthode basée sur la notion de fréquence est fortement discutable. Cependant, l'aspect itératif de notre approche peut être utilisé pour mettre en œuvre une nouvelle itération dans laquelle le concept initial C_0 peut sélectionner un ensemble d'individus plus restreints au sein duquel peut se révéler fréquente une association peu fréquente sur un ensemble plus large d'individus.

Une piste particulièrement intéressante est la mise en évidence des génotypes fortement associés à un phénotype rare. Pour cela, une méthode particulière de recherche d'associations dont le support est faible est la recherche de *règles rares*, *i.e.* d'associations qui contrairement aux règles d'associations, surviennent avec une fréquence inférieure à un seuil défini [SNV07]. Cette expérimentation confirme que les règles rares avec une confiance élevée semblent propices à l'extraction des connaissances en pharmacogénomique où la notion d'intérêt n'est pas forcement couplée à celle de fréquence élevée.

Actuellement seule l'apparition simultanée d'attributs dans une règle est utilisée. On peut supposer à première vue que l'extraction de motifs fréquents (par exemple les motifs fermés fréquents) pourrait être suffisante à l'obtention des mêmes résultats, puisque la notion de règle (et notamment le fait qu'un attribut soit en prémisse ou en conclusion) n'est pas exploitée. Cependant nous utilisons tout d'abord la mesure de la confiance, propre aux règles d'associations, comme une marge permettant à support constant de trouver des associations non systématiques entre les attributs. L'existence de données manquantes ou entachées d'erreurs dans les jeux de données biologiques manipulés est à l'origine de cette considération. Ensuite et surtout, la sémantique associée à une règle, bien qu'encore inexploitée dans la description actuelle de l'AAR, est une des évolutions que nous souhaiterions apporter à cette méthode. Dans ce sens, Rudolph et Völker exploitent par exemple la sémantique des implications entre attributs de la forme $A \rightarrow B$ où A et B sont deux ensembles d'attributs pour définir des nouveaux axiomes en LD de la façon suivante : $\mathtt{A} \sqsubseteq \mathtt{B}$ où $\mathtt{A}$ et $\mathtt{B}$ sont les concepts qui correspondent aux ensembles d'attributs A et B [VR08]. Les résultats très récents présentés par Krötzsch *et al.* sur la description en LD de la sémantique associée à des règles constituent une base solide pour appuyer une telle évolution [MK08].

Il est important de noter que la méthode décrite n'a pas la prétention de remplacer les méthodes statistiques classiques d'analyse de données. En revanche, nous pensons, et l'expérimentation présentée va dans ce sens, que cette méthode peut être utilisée de façon complémentaire, en deuxième approche, pour venir enrichir des résultats initiaux et orienter de nouvelles investigations cliniques ou biologiques.

Des expérimentations supplémentaires non décrites dans le cadre de cette section nous encouragent dans cette direction puisqu'elles permettent d'utiliser les annotations des variants, des gènes, des réseaux métaboliques intégrées à la BC pour mettre en évidence des régularités entre un phénotype intervenant dans une réaction pharmacogénomique et la région particulière de certains gènes (voir règle 5), ou encore des régularités entre un groupe de phénotypes et des variations génomiques localisées sur des gènes impliqués dans une voie métabolique particulière (voir règle 6). Les deux exemples de règles présentés ci-après, obtenus par AAR, illustrent ce genre d'associations.

Règle 5

{isVariantIn_o_interacts_with :Per__-inf-0.08_} =>	*{isDnaVariantIn :intron,* *isVariantIn_o_interacts_with_o_interacts_with :ALOX5,* *isVariantIn_o_interacts_with_o_interacts_with :CYSLTR1}*

Règle 6

{isVariantIn_o_interacts_with :exacerbation} =>	*{isVariantIn_o_interacts_with :arachidonic_acid_metabolism, isVariantIn_o_interacts_with_o_interacts_with :LTC4S, isVariantIn_o_interacts_with_o_interacts_with :eicosanoid_pathway, isVariantIn_o_interacts_with_o_interacts_with :LTA4H}*

L'AAR permet ici l'acquisition et l'insertion de connaissances implicites et nouvelles dans une BC relative à la variabilité de réponses au traitement par montelukast. Ces connaissances sont acquises à partir des résultats d'une investigation particulière et sont définies en tant que telle dans la BC. Une prolongation intéressante de l'utilisation de ce genre de connaissances serait leur interprétation et leur validation expérimentale par des biologistes. Ceci pourraient, sur cette base, généraliser les connaissances mises en évidence par AAR sur un panel restreint, puis les insérer dans la BC avant de les soumettre à PharmGKB.

2.5 Travaux similaires

Pour comparer à l'existant la méthode d'ECBC que nous proposons, il est nécessaire de considérer séparement l'étape préliminaire de peuplement de la BC et la phase d'extraction de connaissances. La première étape de peuplement de l'ontologie *(0)* est abordée dans le Chapitre 2 section 3.2 nous n'y revenons pas ici. Par contre nous distinguerons deux grands groupes de travaux qui manipulent conjointement méthodes de fouille de données et représentation des connaissances.

- Le premier regroupe des travaux sur l'acquisition de connaissances formelles à partir de données, de textes ou de pages Web. Ils ne supposent pas l'existence de connaissances déjà formalisées à l'origine du travail. Ces travaux mènent le processus d'ECBD à son terme où les résultats de la fouille sont interprétés et formalisés dans des langages de représentation des connaissances.
- Le second regroupe des travaux qui tirent parti de connaissances déjà formalisées pour la mise en oeuvre de méthodes d'extraction de connaissances. L'objectif de ce second type de travaux est généralement l'enrichissement des connaissances initialement disponibles.

2.5.1 L'acquisition de connaissances

L'acquisition de connaissances à partir de données, de textes ou de pages Web est également appelée *apprentissage d'ontologie* (traduction de *ontology learning* en anglais) [BCM05]. Les sources de données et les méthodes de fouilles utilisées dans ce cadre sont diverses. Un exemple simple est l'utilisation que font Clerkin *et al.* [CCH01] de l'algorithme COBWEB pour organiser des données selon une hiérarchie de clusters qui est ensuite transformée en une hiérarchie de concepts (ou classes) sous forme d'un graphe RDF reprenant la structure hiérarchique des clusters. La Figure 4.8 représente l'exemple de génération d'ontologie avec COBWEB donnée par Clerkin *et al.*

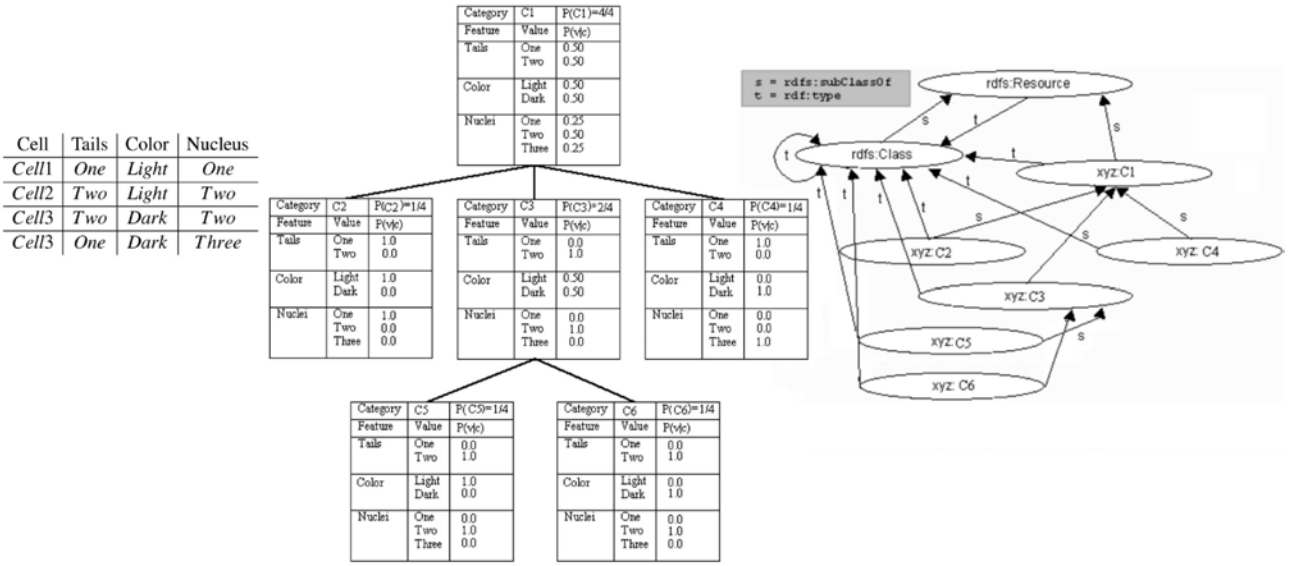

FIG. 4.8 – Un jeu de données exemple concernant la morphologie de cellules soumis à COBWEB, la hiérarchie de cluster produite, et la hiérarchie de concepts (ou classes) RDF déduite [CCH01]

Si le nombre et le volume des bases de données ont considérablement augmenté ces dernières années dans beaucoup de domaines, un volume considérable de connaissances n'est encore disponible que sous forme de texte en langage naturel, et notamment d'articles de revues spécialisées. En conséquence de nombreux travaux ce sont intéressés à extraire et formaliser des connaissances contenues dans des corpus

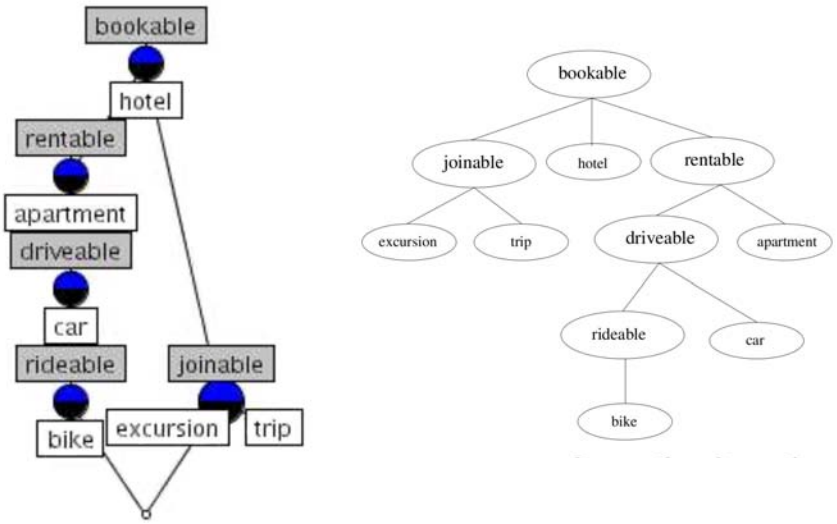

FIG. 4.9 – Un treillis de concepts, notation réduite, produit à partir de textes (à gauche) et la hiérarchie de concepts en laquelle il est transformé (à droite) suivant la méthode proposée dans [CHS05]

de textes (voir [BCM05] pour une vue d'ensemble). Dans cette optique, Cimiano *et al.* dérivent à partir de textes une hiérarchie de concepts [CHS05]. Pour cela ils construisent dans un premier temps un contexte formel à l'aide de méthodes de TAL qui leur permettent d'extraire des relations entre termes associés dans les textes. Dans un deuxième temps, le contexte formel construit est soumis à des méthodes d'ACF pour produire un treillis ensuite transformé et compacté en un ordre partiel spécifique qui constitue une ontologie. Un exemple de treillis produit et la hiérarchie de concepts en laquelle il est transformé sont représentés Figure 4.9. La transformation du treillis revient à *(1)* retirer le concept *bottom*, *(2)* créer un concept dit *ontologique* pour chaque concept formel avec comme nom l'*intension* du concept formel, et *(3)* créer un sous-concept relié au précédent pour chaque élément présent dans l'*extension* du concept formel en question. La hiérarchie produite est finalement réduite afin de limiter le nombre potentiellement très élevé de concepts qui résultent de la transformation d'un grand treillis. Pour cela les concepts dit *ontologiques* qui ont la même *extension* en terme de concepts terminaux que leurs sous-concepts (*i.e.* les mêmes nœuds feuilles dans la hiérarchie) sont supprimés. Dans l'exemple représenté Figure 4.9, le concept *rideable* serait de cette façon supprimé de la hiérarchie.

Bendaoud *et al.* ont proposé plus récemment une méthode d'acquisition de connaissances à partir de textes qui s'appuie sur l'ACF. Celle ci présente deux avantages principaux par rapport à [BCM05]. Le premier est de produire non seulement une hiérarchie de concepts, mais également des instances associées aux concepts. Ici le concept *bottom* est éliminé et les éléments de l'extension d'un concept formel servent également à décrire en LD les concepts de la hiérarchie comme dans [BCM05]. En revanche les éléments de l'intension des concepts formels servent à la création d'individus qui instancient le concept correspondant à l'élément décrit en extension. La Figure 4.10 illustre cette transformation d'un treillis en une hiérarchie de concepts plus instances ; elle peut être comparée à la Figure 4.9. Dans un sens la façon de décrire en LD les concepts formels dépend de la façon dont les connaissances contenues dans les textes est codée dans le contexte formel. Cependant les correspondances entre d'abord l'extension de concepts formels et la description de concepts en LD et ensuite entre intension et instances semblent relativement naturelles. Le deuxième avantage de cette méthode consiste en l'enrichissement de la hiérar-

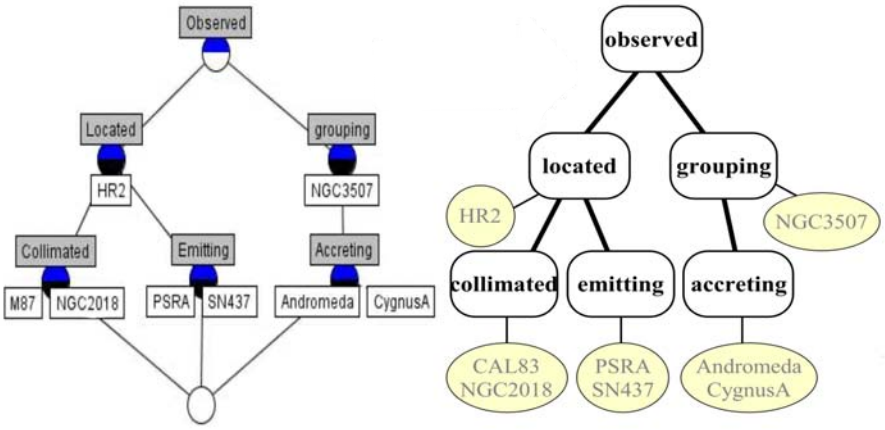

FIG. 4.10 – Un treillis de concepts, notation réduite, produit à partir de textes (à gauche) et la hiérarchie de concepts instanciée en laquelle il est transformé (à droite) suivant l'alternative proposée dans [BTN08]

.

chie de concepts par des rôles entre les concepts en utilisant une méthode formelle appelée l'Analyse de Relationnelle de Concepts ou ARC. Cet enrichissement présente la particularité de prendre en compte la hiérarchie des concepts pour la généralisation de relations extraites des textes.

Les connaissances sur la structuration des documents analysés peuvent également servir à guider le processus d'acquisition de connaissance. C'est notamment le cas dans [KAB06] où la connaissance d'experts sur la structuration des documents HTML est utilisée pour favoriser la construction de clusters de termes homogènes à partir de pages Web.

Cette catégorie de travaux, rassemblée sous le titre *acquisition de connaissances* peut être comparée à l'enchaînement des étapes *(ii)* et *(iii)* de notre méthode d'ECBC. Ces travaux permettent la formalisation de connaissances contenues dans des données, des textes ou des pages Web et ce à l'aide de méthodes de fouille de données et éventuellement de représentations des connaissances du domaine pour [KAB06, BTN08]. Dans tous les cas, les connaissances acquises par le processus sont enregistrées mais ne sont pas considérées de sorte à pouvoir être réutilisées dans les opérations d'acquisition de connaissances suivantes. Finalement ces méthodes considèrent de façon séparée sources de données (bases de données, textes, pages Web) et représentation des connaissances du domaine sans prendre en compte l'existence des ontologies du domaine existant.

2.5.2 La manipulation de connaissances pour extraire de nouvelles connaissances

Utilisation des mécanismes de raisonnement comme méthode d'apprentissage Un moyen original de se servir de connaissances formalisées pour l'extraction de connaissances potentiellement utiles est l'utilisation de mécanismes de raisonnement comme méthode d'apprentissage sur une BC , plutôt que d'algorithmes de fouille sur des bases de données. Ce sont alors les concepts, rôles, instances et axiomes de la BC qui sont directement manipulés par ces mécanismes de raisonnement. En pratique, ceux-ci sont appliqués à des BC en LD pour formaliser explicitement des connaissances implicites. Le plus souvent ils mettent en lumière des éléments de connaissance évidents pour l'analyste (humain) et sont rarement efficaces pour la découverte de connaissances dans le cadre de l'ECBD. Un travail qui fait exception est la *classification des protéines phosphatases* proposée par Wostencroft *et al.* [WLT+06]. Les auteurs se basent sur une ontologie en LD qui décrit la composition en domaines des protéines de la famille des phos-

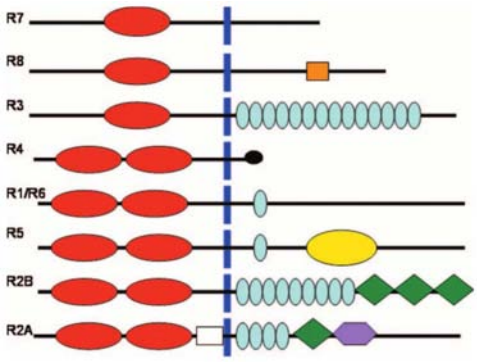

FIG. 4.11 – Les différences d'organisation des domaines dans une sous-famille de protéines phosphatases : les *récepteurs tyrosines phosphatases*. Ces organisations sont représentées dans l'ontologie des phosphatases et utilisées pour la classification automatique de nouvelles protéines [WLT+06].

phatases (voir Figure 4.11). Une protéine y est représentée comme un individu instance de l'ontologie auquel est associé la description de sa composition en domaines protéiques. Ils appliquent alors sur la BC associée un mécanisme de raisonnement classique de classification d'instances. La comparaison d'une protéine (donc d'un individu) aux descriptions des familles des phosphatases (*i.e.* des concepts définis) permet ainsi sa classification automatique dans la famille à laquelle elle appartient. La classification résultante a permis d'affiner la classification des phosphatases définie par les experts et de corriger pour certaines protéines l'annotation qui indique leur appartenance à une famille ou à une autre.

Moyennant quelques adaptations, il est également possible de transformer (ou coder) certains éléments de connaissance afin que ceux ci puissent-être considérés par des méthodes de fouille de données.

Fouille de bases de cas Les éléments de connaissances fouillées par le système CABAMAKA présenté dans [dBL+07] sont des *cas*, *i.e.* des couples (problème, solution), regroupés dans une *base de cas*. La fouille de la base de cas, par l'extraction des motifs fermés fréquents, a pour objectif la découverte de *connaissances d'adaptation i.e.* un élément de connaissance qui décrit comment évolue la solution entre deux couples (problème, solution) quand le problème évolue. Les résultats extraits prennent la forme de motifs fermés fréquents dont chaque élément décrit l'adaptation d'une propriété et dont l'ensemble permet de déduire des connaissances d'adaptation potentiellement utiles. [dBL+07] propose, dans le contexte de bases de cas de traitement du cancer du sein, un exemple de connaissance d'adaptation extraite et son interprétation.

Clustering conceptuel Les éléments de connaissance peuvent également être les individus d'une BC en LD comme dans [FdE08] pour lequel les auteurs proposent une méthode de clustering hiérarchique basée sur une distance sémantique mesurée entre individus. En accord avec cette distance, deux individus qui appartiennent à un grand nombre de concepts communs dans la BC sont proches, et inversement deux individus qui ne peuvent pas appartenir à des concepts communs sont éloignés (voir [FdE08] pour la définition formelle de la distance). Le calcul de la distance nécessite l'intervention de mécanismes de raisonnement pour déterminer l'instanciation des concepts (*instance checking*). Une méthode de clustering inspirée des K-plus proches voisins utilise ensuite les distances entre individus pour construire une hiérarchie de clusters. Les clusters construits peuvent être annotés avec une description en LD qui caractérise au mieux les individus présents dans le cluster en question tout en discriminant les individus des clusters de même niveau dans la hiérarchie. Cette description correspond au *subsumeur commun le plus*

spécifique de l'ensemble des *concepts les plus spécifiques* de chaque individu du cluster. La description résultante est une approximation et demeure dépendante de la description des concepts et de la répartition des individus dans la BC (*i.e.* de la conceptualisation). L'ajout de nouveaux individus dans la BC induit des modifications dans la structure hiérarchique en terme soit de nouveau cluster, soit de nouvelle description d'un cluster existant. Cependant la modification de la *TBox* initiale à la vue de clusters et de leur description en LD associée n'est pas considérée.

Ces deux dernières approches manipulent des connaissances représentées formellement dans une LD, mais ne réutilisent pas (ou partiellement) les résultats de fouille pour raffiner ou enrichir la BC analysée.

En revanche, les travaux théoriques de Rudolph *et al.* [Rud06] et de Baader *et al.* [BGSS07] (présentés dans la section 2.2), et notre méthode d'AAR (section 2.3) autorisent l'enrichissement de la BC initiale.

REMARQUE : Le travail de Ferré *et al.* est à noter également. Ils utilisent aussi LD et ACF conjointement mais dans un objectif inverse au nôtre [FRS05]. L'objectif n'est pas d'enrichir une ontologie (ou une BC) à partir de résultats d'ACF mais d'enrichir un treillis de concepts à l'aide de la sémantique associée aux LD.

3 Discussion

Cette section discute dans un premier temps la méthode d'Analyse des Assertions de Rôles (AAR) présentée dans la section 2.3 puis, dans un second temps la proposition plus générale d'Extraction de Connaissances *à partir* de Base de Connaissances (ECBC).

L'AAR s'appuie sur un mécanisme qui transforme des connaissances représentées en LD sous la forme d'un contexte formel manipulable par les méthodes d'ACF. Cette transformation permet de représenter dans le contexte formel produit *(1)* des assertions de rôles qui relient les individus dans un graphe d'assertions, *(2)* les assertions de concepts représentées par les attributs du contexte. Ainsi cette représentation des connaissances est une représentation partielle du contenu de la BC. Par exemple, les relations de subsomption entre concepts et entre rôles ne sont pas représentées dans le contexte. Il peut pourtant apparaître intéressant de les considérer dans l'idée de retranscrire de façon plus complète les connaissances contenues dans la BC.

Une méthode alternative à l'exploration des graphes d'assertions, pour retranscrire de façon systématique les connaissances d'une BC dans un format manipulable par des méthodes de fouille, pourrait être l'utilisation de la méthode d'Analyse Relationnelle de Concept (ARC) [HHNV07]. Cette méthode peut être utilisée pour considérer l'ensemble des connaissances d'une BC à condition de retranscrire celles-ci dans plusieurs contextes formels (par exemple un contexte pour les assertions de concepts, un autre pour les assertions de rôles, encore un autre pour la hiérarchie de concepts, etc.). L'ARC permet ensuite, sur la base des régularités présentes dans les contextes, de générer de nouvelles descriptions de concepts en LD qui prennent en considération les relations entre les éléments impliqués dans différents contextes. La contrainte importante associée à l'utilisation de l'ARC est la définition manuelle des contextes, de leurs relations, et enfin l'interprétation des nombreux concepts produits automatiquement. En comparaison avec l'ARC, notre méthode d'AAR propose un compromis qui permet une retranscription partielle des connaissances de la BC et la production d'un contexte de taille raisonnable. Une des hypothèses de départ de l'AAR est notamment que les assertions de rôles directes et indirectes peuvent décrire des relations plus ou moins fréquentes entre différentes catégories d'individus, et par ce biais sont des connaissances susceptibles d'être porteuses de connaissances nouvelles.

Un moyen d'évaluer concrètement l'apport de l'AAR serait de comparer les résultats d'une recherche des $\mathcal{RMNR}$ sur un jeu de données brut, par exemple les résultats de l'investigation clinique du montelukast utilisée pour l'expérimentation présentée section 2.4, avec les règles et résultats produits par l'AAR. Cette expérimentation pourrait permettre de justifier l'effort que nécessite la construction de l'ontologie et la définition des mappings données-assertions pour la découverte de connaissances.

Une utilisation intéressante de l'AAR pourrait consister à positionner le concept initial C_0 à la racine de la *TBox* ($C_0 \equiv \top$), puis à recueillir la totalité des nouveaux concepts dérivés des $\mathcal{RMNR}$ dans une nouvelle *TBox* vierge. La comparaison entre la hiérarchie de concepts obtenue après classification de cette nouvelle *TBox* et la hiérarchie de concepts de la BC initiale permettrait d'évaluer en un sens la capacité des assertions de rôles d'une BC à refléter la représentation des connaissances établies et à en proposer de nouvelles. Dans une idée proche les méthodes d'évaluation d'ontologies pourraient permettre de mesurer la progression entre la BC initiale et la BC finale, après AAR [GCCL06].

L'AAR (détaillée et expérimentée sections 2.3 et 2.4) se veut une illustration de la proposition plus générale d'ECBC. En effet l'ECBC peut être mise en œuvre à l'aide de différentes méthodes. Par exemple des méthodes de fouille de données alternatives à la recherche des $\mathcal{RMNR}$ pourraient être utilisées. En fonction de la méthode choisie, il serait alors indispensable d'adapter les étapes de préparation et d'interprétation des résultats de fouille qui dépendent respectivement des formats d'entrée et du type d'unités d'information produit par la fouille. Des méthodes de fouille alternatives qu'il semble pertinent

d'expérimenter sont par exemple la construction d'arbres de décision ou le clustering conceptuel. Un système de représentation des connaissances différent des LD pourrait également être envisagé.

Suivant l'exemple de l'AAR, le processus d'extraction de connaissances reste centré sur un algorithme de fouille de données qui prend en entrée des connaissances préalablement préparées en un format compatible et produit en sortie des unités d'information extraites à interpréter, formaliser, et insérer dans la BC. Malgré différents essais, cela reste un défi de considérer des méthodes d'apprentissage capables de manipuler les axiomes d'une BC, *i.e.* capables de tenir compte de la sémantique qui leur est associée et de leur régularités [Mug91, Nap92, Ser07].

Conclusion et perspectives

L'ensemble de la thèse présentée dans ce manuscrit conforte la conviction selon laquelle l'Extraction de Connaissances *à partir* de Bases de Données (ECBD) doit être guidée à partir des connaissances du domaine. Dans ce contexte, nous avons proposé et mis en œuvre un ensemble cohérent de méthodes afin de formaliser les connaissances d'un domaine, intégrer des données hétérogènes relatives à ce domaine au sein d'une Base de Connaissances (BC), et enfin analyser le contenu de cette BC pour en extraire de nouvelles connaissances.

La première contribution décrit une méthode de construction d'ontologie qui favorise la réutilisation d'ontologies déjà existantes en les intégrant au processus de construction. De plus, cette méthode inclut l'évaluation de la capacité de l'ontologie construite à représenter les connaissances du domaine.

Ensuite, nous décrivons une approche d'intégration de données qui s'appuie sur une formalisation théorique de la mise en correspondance de données et de connaissances. La mise en œuvre de cette approche conduit à associer une sémantique préalablement définie à des données hétérogènes afin de les intégrer au sein d'une BC.

La BC résultante constitue l'élément central du processus d'Extraction de Connaissances *à partir* d'une Base de Connaissances (ECBC). L'idée originale ici est d'explorer les régularités d'une BC par opposition à la recherche de régularités à partir de bases de données. Nous rapportons des résultats prometteurs sur la définition et la mise en œuvre d'une méthode d'ECBC particulière appelée l'Analyse d'Assertions de Rôles (AAR). Cette méthode s'attache à explorer les régularités dans les assertions d'une BC. Les unités de connaissances produites sont exprimées suivant le même formalisme que celui de la BC ce qui permet, ensuite, leur manipulation par des mécanismes de raisonnement en vue de leur intégration cohérente dans la BC. Cette dernière contribution propose ainsi un moyen de mettre en œuvre un processus d'Extraction de Connaissance *guidée par* les Connaissances du Domaine.

Ces résultats illustrent que l'ECBC peut être employée avec succès pour la découverte de connaissances en pharmacogénomique. De plus le cadre général de l'ECBC peut être réutilisé pour décrire de nouvelles de méthodes différentes de la nôtre.

Par ailleurs, l'ensemble cohérent des méthodes décrites dans la thèse a été appliqué au domaine de la pharmacogénomique. Nous avons ainsi construit deux ontologies de domaine. La première, SNP-Ontology, formalise les connaissances sur les variations génomiques. La seconde, SO-Pharm, formalise les connaissances du domaine de la pharmacogénomique. SNP-Ontology et SO-Pharm ont reçu un accueil favorable par la communauté scientifique intéressée par la mise à disposition et le partage des bio-ontologies.

Nous avons ensuite employé ces ontologies pour intégrer, selon l'approche d'intégration proposée, des données pharmacogénomiques issues de bases de données hétérogènes. Le résultat de cette intégrations est le peuplement de Bases de Connaissances relatives à ce domaine particulièrement concerné par le problème de l'intégration de données. Dans le cas de données relatives au génotype, notre système SNP-Converter permet de réconcilier les différentes représentations des variations génomiques. Dans le cas d'investigations cliniques, notre système iSO-Pharm permet de cumuler des données relatives au génotype, au phénotype et au médicament provenant d'investigations réelles.

Enfin nous avons expérimenté notre méthode d'ECBC, l'AAR, dans le contexte d'une BC instanciée par iSO-Pharm à partir d'une investigation clinique particulière et de données complémentaires provenant de bases de données publiques. Nos résultats confortent et enrichissent les résultats publiés sur cet essai qui ont été obtenus grâce aux méthodes statistiques classiques pour ce genre d'études. De plus, l'interprétation des résultats de la fouille a permis d'insérer de nouvelles connaissances dans la BC initiale.

Les perspectives suscitées par ce travail sont nombreuses. En premier lieu il conviendrait de définir une procédure d'évaluation de la pertinence et de la nouveauté des unités de connaissances découvertes par l'approche d'ECBC proposée dans cette thèse. Une telle procédure pourrait s'appuyer sur des mesures comparant les unités de connaissances au contenu préalable de la BC pour aider l'expert dans la tâche de validation des résultats de fouille. Une deuxième perspective, plus vaste, porte sur l'élargissement du domaine d'application considéré, *i.e.* la pharmacogénomique, à l'étude plus globale des *interactions gènes-environnement*. En effet, les traitements médicamenteux et les facteurs génétiques ne sont pas les seuls éléments ayant un impact sur les traits du phénotype. Par exemple l'état nutritionnel, les micro-organismes intestinaux, les maladies vécues, l'âge d'un individu ont des impacts sur son phénotype, et donc sur sa réponse à un médicament. Dans cette direction il est possible d'étendre les connaissances représentées dans SO-Pharm pour considérer les influences possibles de nouveaux facteurs. Il serait alors possible d'envisager de peupler la version étendue de SO-Pharm à partir des résultats d'essais ou de cohortes étudiant de façon plus générales ces interactions gènes-environnement [Mau06, CLC+06, ORT08, RLSN08]. La base de connaissances résultante pourrait être utilisée, suivant l'approche d'ECBC proposée dans cette thèse, avec comme objectif d'analyser simultanément les influences quantitatives et qualitatives des divers facteurs sur le phénotype. Suivant cette idée il serait intéressant d'utiliser des méthodes de fouilles de données alternatives à celles expérimentées dans le cadre de l'AAR. Par exemple, l'extraction de motifs rares, la classification hiérarchique, la fouille de données temporelles sont des méthodes qui paraissent adaptées aux particularités du domaine. L'utilisation des connaissances à priori pour la découverte de connaissances sur les interactions gènes-environnement repose également sur la formalisation de la notion de phénotype, de ces différents niveaux de granularité, de ces variations inter-individuelles et temporelles. Une troisième perspective porte sur l'appropriation par les experts des outils de gestion des connaissances. Les nombreux travaux exploitant les technologies du Web sémantiques laissent présager l'émergence de systèmes permettant de compléter et d'interroger une BC via un environnement de visualisation intuitif qui exploite par exemple la structure en graphe inhérente à une BC implémentée en OWL ou encore les capacités d'interrogation du langage SPARQL [RMKM08, BdLM08]. L'intégration de ces avancées au niveau d'une interface utilisateur permettrait à l'expert d'être plus autonome tout au long du processus d'ECBC. Enfin, une quatrième perspective est l'application de l'approche d'ECBC à des domaines différents de celui de la pharmacogénomique. De telles expérimentations permettraient d'évaluer à quel point notre approche est générique.

Au terme de ce travail, il apparaît que les contributions apportées tant au domaine de l'informatique qu'à celui de la pharmacogénomique sont fructueuses et prometteuses. Le caractère particulièrement interdisciplinaire de cette thèse nous a conduit au développement de systèmes opérationnels pour la pharmacogénomique à partir de technologies du Web sémantique. Cependant le rôle de la pharmacogénomique dans ce travail de thèse ne se limite pas à un simple domaine d'application. Par la complexité de ses problématiques, ce domaine a orienté et motivé les approches informatiques proposées qui resteront applicables à d'autres domaines. Nous espérons ainsi avoir contribué à la concrétisation du Web sémantique en dépassant le niveau du Web de données (en anglais *Web of data*).

Annexe A

Algorithme de recherche des $\mathcal{RMN}$ et des $\mathcal{RMNR}$

Description

Cet algorithme permet la recherche des Règles Minimales Non-redondantes ($\mathcal{RMN}$) et des Règles Minimales Non-Redondantes Réduites ($\mathcal{RMNR}$) [Sza06]. Pour cela, il prend en entrée trois paramètres : *(1)* les motifs fermés fréquents (MFF), *(2)* les générateurs fréquents (GF), et *(3)* les valeurs du support des motifs. Pour décrire cet algorithme nous utilisons deux fonctions :

`getProperSuperSet` prend deux paramètres : un ensemble S de motifs et un motif p. La fonction retourne le super motif direct de p dans S. Cette fonction peut tirer parti d'une structuration des motifs sous forme d'un treillis.

`getSupportOf` prend en paramètre un motif quelconque et renvoie son support. Cette fonction peut également tirer parti de l'organisation en treillis des motifs.

REMARQUE : L'algorithme peut être facilement adapté pour isoler les ensembles de règles qui correspondent à la Base Générique ($\mathcal{BG}$), la Base Informative ($\mathcal{BI}$), et la Base Informative Réduite ($\mathcal{BIR}$). [Sza06]

Algorithme

Algorithme A.1 Recherche des $\mathcal{RMN}$ et des $\mathcal{RMNR}$

Entrée : MFF, GF, valeurs de support des motifs
Sortie : ensemble des $\mathcal{RMN}$ et des $\mathcal{RMNR}$

```
1 : Pour chaque générateur g
2 :   C_G ← getProperSuperSet(MFF,g)      {retourne le super motif de g parmi les MFF}
3 :
4 :   Pour chaque élement c de C_G
5 :     premisse ← g
6 :     conclusion ← (c\g)
7 :     supportPremisse ← g.support
8 :     supportConclusion ← getSupportOf(conclusion)
9 :
10 :    regle ← (premisse → conclusion)
11 :
12 :    si conf(regle) ≥ min_conf                {conf(regle)=supp(c/supp(g))}
13 :    {L'étape suivante est optionnelle. Elle doit être exécutée
14 :    si l'on souhaite extraire les RMNR au lieu des RMN}
15 :      si conf(regle) ≠ 1.0
16 :        C_G ← C_G\getProperSuperSet(MFF,c) {le supermotif de c est éliminé de C_G}
17 :      fin si
18 :      R ← R ∪ {regle}
19 :    fin si
20 :  Pour chaque
22 : Pour chaque
23 : Retourner R
```

Annexe B

Constructeurs en LD

Nom du constructeur	syntaxe en LD	syntaxe OWL	sémantique associée
Concept	$\mathtt{C}$	C (URI)	$\mathtt{C}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
Concept universel	$\top$	`owl:Thing`	$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$
Bottom	$\bot$	`owl:Nothing`	$\bot^{\mathcal{I}} = \emptyset$
Intersection	$\mathtt{C} \sqcap \mathtt{D}$	`intersectionOf`(C D)	$(\mathtt{C} \sqcap \mathtt{D})^{\mathcal{I}} = \mathtt{C}^{\mathcal{I}} \cap \mathtt{D}^{\mathcal{I}}$
Union	$\mathtt{C} \sqcup \mathtt{D}$	`unionOf`(C D)	$(\mathtt{C} \sqcup \mathtt{D})^{\mathcal{I}} = \mathtt{C}^{\mathcal{I}} \cup \mathtt{D}^{\mathcal{I}}$
Négation	$\neg\mathtt{C}$	`complementOf`(C)	$(\neg\mathtt{C})^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus \mathtt{C}^{\mathcal{I}}$
Enumération	$\{\mathtt{a}, \mathtt{b} \ldots\}$	`oneOf`(a b ...)	$\{\mathtt{a}, \mathtt{b} \ldots\}^{\mathcal{I}} = \{\mathtt{a}^{\mathcal{I}}, \mathtt{b}^{\mathcal{I}} \ldots\}$
Quantificateur existentiel	$\exists\mathtt{R}.\mathtt{C}$	`restriction`(R `someValuesFrom`(C))	$(\exists\mathtt{R}.\mathtt{C})^{\mathcal{I}} = \{x \mid \exists y.\ (x,y) \in \mathtt{R}^{\mathcal{I}} \wedge y \in \mathtt{C}^{\mathcal{I}}\}$
Quantificateur universel	$\forall\mathtt{R}.\mathtt{C}$	`restriction`(R `allValuesFrom`(C))	$(\forall\mathtt{R}.\mathtt{C})^{\mathcal{I}} = \{x \mid \forall y.\ (x,y) \in \mathtt{R}^{\mathcal{I}} \rightarrow y \in \mathtt{C}^{\mathcal{I}}\}$
Restriction à une valeur	$\ni \mathtt{R}.\mathtt{a}$ *ou* $\mathtt{R}.\{\mathtt{a}\}$	`restriction`(R `hasValue`(a))	$(\ni \mathtt{R}.\mathtt{a})^{\mathcal{I}} = \{x \mid (x, a^{\mathcal{I}}) \in \mathtt{R}^{\mathcal{I}}\}$
Restrictions non qualifiées de cardinalité	$= n\ \mathtt{R}$	`restriction`(R `cardinality`(C))	$(= n\ \mathtt{R})^{\mathcal{I}} = \{x \mid \mathrm{card}\{y \mid (x,y) \in \mathtt{R}^{\mathcal{I}}\} = n\}$
	$\geqslant n\ \mathtt{R}$	`restriction`(R `minCardinality`(C))	$(\geqslant n\ \mathtt{R})^{\mathcal{I}} = \{x \mid \mathrm{card}\{y \mid (x,y) \in \mathtt{R}^{\mathcal{I}}\} \geqslant n\}$
	$\leqslant n\ \mathtt{R}$	`restriction`(R `maxCardinality`(C))	$(\leqslant n\ \mathtt{R})^{\mathcal{I}} = \{x \mid \mathrm{card}\{y \mid (x,y) \in \mathtt{R}^{\mathcal{I}}\} \leqslant n\}$
Quantificateur existentiel	$\exists\mathtt{S}.\mathtt{T}$	`restriction`(S `someValuesFrom`(T))	$(\exists\mathtt{S}.\mathtt{T})^{\mathcal{I}} = \{x \mid \exists y.\ (x,y) \in \mathtt{S}^{\mathcal{I}} \wedge\ y \in \mathtt{T}^{\mathcal{I}}\}$
Quantificateur universel	$\forall\mathtt{S}.\mathtt{T}$	`restriction`(S `allValuesFrom`(T))	$(\forall\mathtt{S}.\mathtt{T})^{\mathcal{I}} = \{x \mid \forall y.\ (x,y) \in \mathtt{S}^{\mathcal{I}} \rightarrow y \in \mathtt{T}^{\mathcal{I}}\}$
Restriction à une valeur	$\ni \mathtt{S}.\mathtt{a}$ *ou* $\mathtt{S}.\{\mathtt{a}\}$	`restriction`(S `hasValue`(a))	$(\ni \mathtt{S}.\mathtt{a})^{\mathcal{I}} = \{x \mid (x, a\mathtt{D}) \in \mathtt{S}^{\mathcal{I}}\}$
Restriction non qualifiée de cardinalité	$= n\ \mathtt{S}$	`restriction`(S `cardinality`(T))	$(= n\ \mathtt{S})^{\mathcal{I}} = \{x \mid \mathrm{card}\{y \mid (x,y) \in \mathtt{S}^{\mathcal{I}}\} = n\}$
	$\geqslant n\ \mathtt{S}$	`restriction`(S `minCardinality`(T))	$(\geqslant n\ \mathtt{S})^{\mathcal{I}} = \{x \mid \mathrm{card}\{y \mid (x,y) \in \mathtt{S}^{\mathcal{I}}\} \geqslant n\}$
	$\leqslant n\ \mathtt{S}$	`restriction`(S `maxCardinality`(T))	$(\leqslant n\ \mathtt{S})^{\mathcal{I}} = \{x \mid \mathrm{card}\{y \mid (x,y) \in \mathtt{S}^{\mathcal{I}}\} \leqslant n\}$

TAB. B.1 – Constructeurs de concepts en Logique de Descriptions LD et leurs correspondances en OWL. C et D sont des concepts (respectivement C et D sont des classes), T est un concept particulier qui correspond à un type de données (un *Datatype* en OWL), n est un nombre, a et b sont des individus, R un rôle (une propriété d'objet ou *ObjectProperty* en OWL) et S un rôle dont le co-domaine correspond à un concept de même type que T (une propriété de données ou *DatatypeProperty* en OWL).

nom du constructeur	syntaxe en LD	syntaxe abstraite OWL	sémantique associée
Rôle inverse	$\mathtt{R}^-$	`inverseOf`(R)	$(\mathtt{R}^-)^{\mathcal{I}} = \{(x,y) \mid (y,x) \in \mathtt{R}^{\mathcal{I}}\}$
Composition de rôle	$\mathtt{R} \circ \mathtt{Q}$	–	$(\mathtt{R} \circ \mathtt{Q})^{\mathcal{I}} = \{(x,z) \mid \exists y.(x,y) \in \mathtt{R}^{\mathcal{I}} \wedge (y,z) \in \mathtt{Q}^{\mathcal{I}}\}$

TAB. B.2 – Constructeurs de rôles en Logique de Descriptions LD et leurs correspondances en OWL. R et Q sont des rôles (des propriétés d'objet ou *ObjectProperty* en OWL)

Annexe C

Exemple de code OWL

```
<?xml version="1.0"?>

<rdf:RDF xmlns="http://www.loria.fr/~coulet/exemple_de_bc.owl#"
    xml:base="http://www.loria.fr/~coulet/exemple_de_bc.owl"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
   <owl:Ontology rdf:about=""/>

   <owl:ObjectProperty rdf:ID="estTrait&#233;">
      <rdfs:domain rdf:resource="#Patient"/>
      <rdfs:range rdf:resource="#TraitementM&#233;dicamenteux"/>
   </owl:ObjectProperty>
   <owl:Class rdf:ID="Patient">
      <rdfs:subClassOf rdf:resource="#Person"/>
   </owl:Class>
   <owl:Class rdf:ID="PatientSousTraitement">
      <owl:equivalentClass>
         <owl:Class>
            <owl:intersectionOf rdf:parseType="Collection">
               <owl:Restriction>
                  <owl:onProperty rdf:resource="#estTrait&#233;"/>
                  <owl:someValuesFrom rdf:resource="#TraitementM&#233;dicamenteux"/>
               </owl:Restriction>
               <owl:Class rdf:about="#Patient"/>
            </owl:intersectionOf>
         </owl:Class>
      </owl:equivalentClass>
      <rdfs:subClassOf rdf:resource="&owl;Thing"/>
   </owl:Class>
   <owl:Class rdf:ID="Person"/>
   <owl:Class rdf:ID="TraitementM&#233;dicamenteux"/>

 <Patient rdf:ID="adrien">
      <estTraité rdf:resource="#cureDAntibiotique"/>
   </Patient>
   <TraitementMédicamenteux rdf:ID="cureDAntibiotique"/>

</rdf:RDF>
```

FIG. C.1 – Code OWL qui correspond à la BC représentée dans le Tableau 2.4. Ce code est enregistré dans le fichier “exemple_de_bc.owl”.

Annexe D

Modèle conceptuel de SO-Pharm

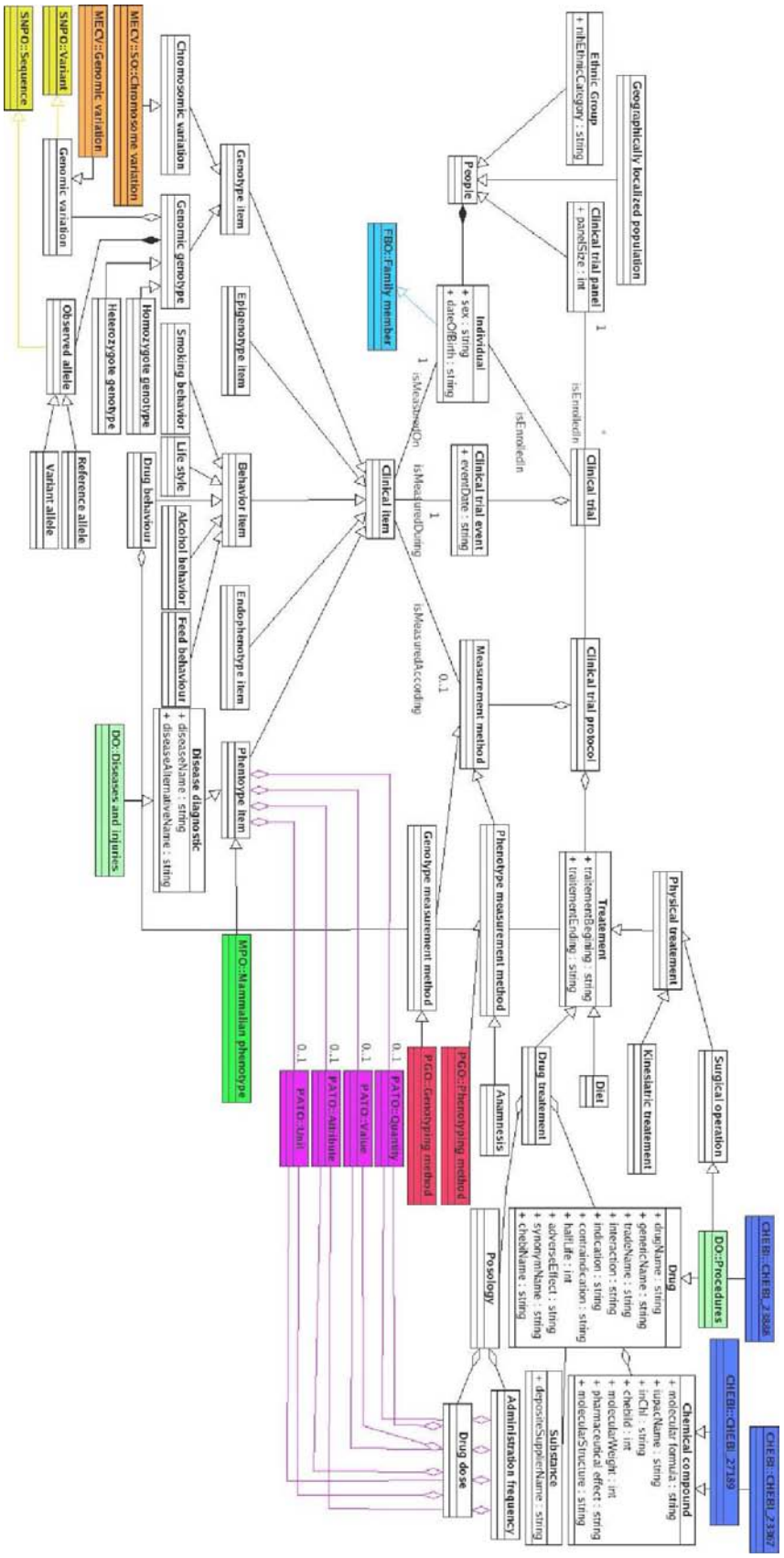

FIG. D.1 – Diagramme de classes UML donnant une vue générale, mais partielle, de la conceptualisation de SO-Pharm

Annexe E

Captures d’écrans de SNP-Converter

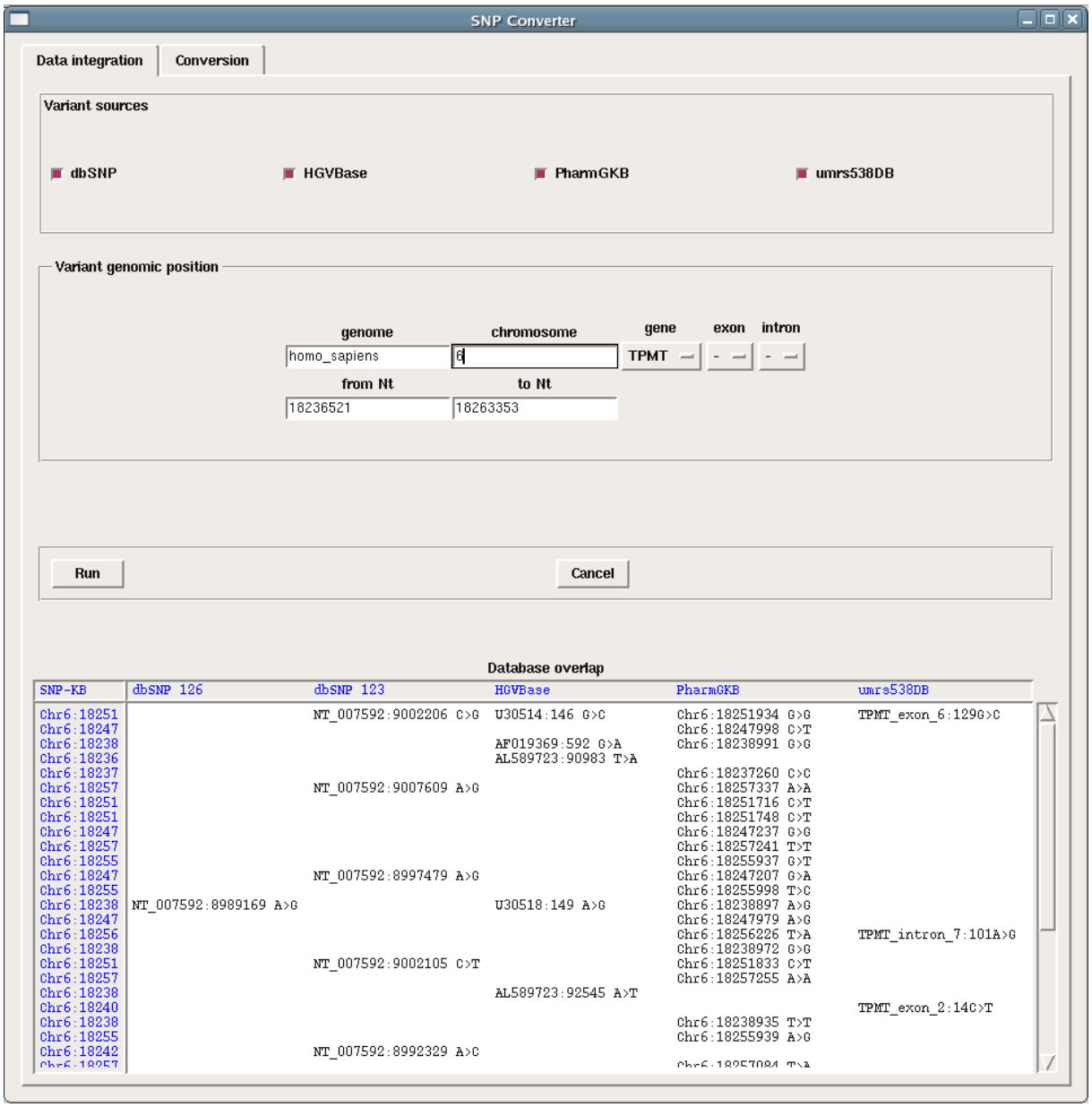

FIG. E.1 – Capture d'écran de SNP-Converter. L'onglet présenté s'intitule *Data integration*. Il propose de sélectionner une liste de sources de données et une portion du génome : un exon, un intron, un gène entier ou un espace situé entre deux nucléotides. L'exécution de la fonction d'intégration de données de SNP-Converter par le bouton *Run* permet l'instanciation d'une Base de Connaissances SNP-KB qui permet d'évaluer le recouvrement des données contenues dans les différentes sources et représentées dans le cadre intitulé *Database overlapp*. Par exemple, le premier variant de la liste est initialement présent dans les 4 sources de données sélectionnées, le second est présent uniquement dans PharmGKB, le troisième est dans HGVBase et PharmGKB.

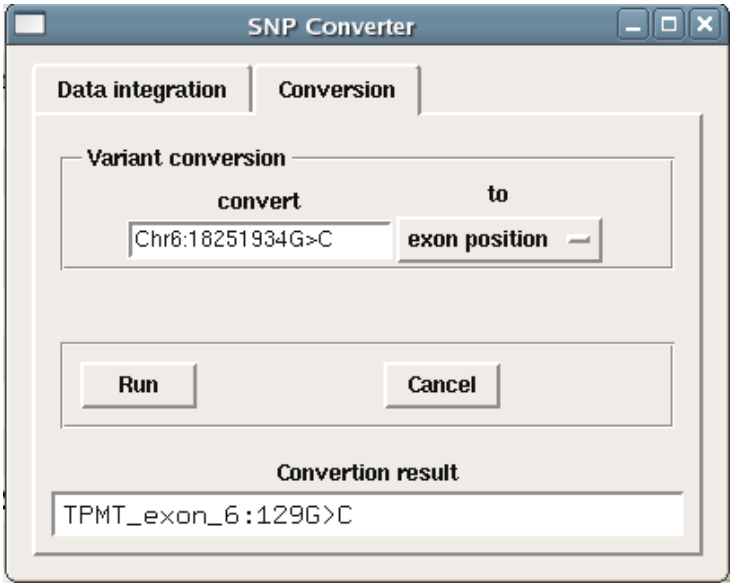

FIG. E.2 – Capture d'écran de SNP-Converter. L'onglet présenté s'intitule *Conversion*. Il propose de saisir la descritpion d'un variant, ici Chr6 :18251934G>C, et de choisir un type de description différent pour décrire le variant, ici la position par rapport à l'exon. L'exécution par le biais du bouton *Run* construit la description du variant donnée selon la description demandée : TPMT_exon_6 :129G>C. Le variant donné en entrée peut être soit un identifiant d'une base de données, soit être décrit suivant la nomenclature HGVS.

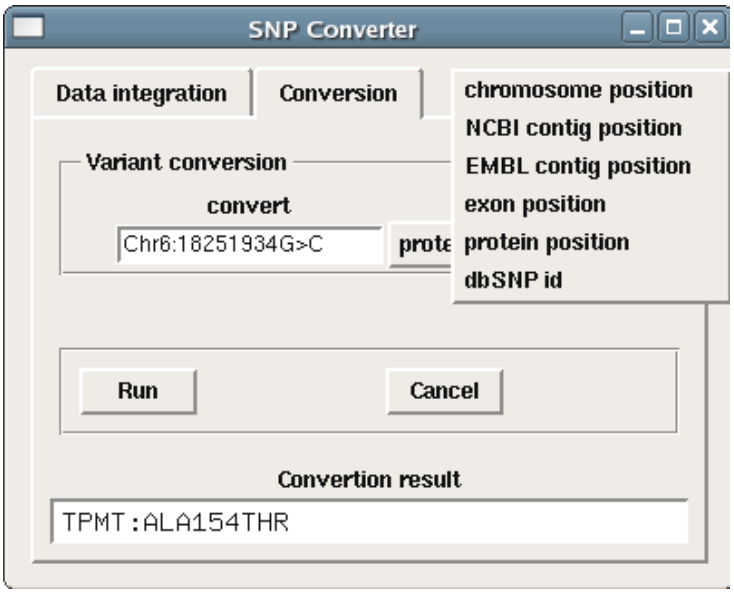

FIG. E.3 – Capture d'écran de SNP-Converter. L'onglet est le même que celui présenté dans la Figure E.2. Cette figure représente en plus les différents type de description suivant lesquelles il est possible de convertir le variant donné : nomenclature HGVS du variant positionné relativement à la séquence du chromosome, de contigs, de l'exon, de l'intron, de la protéine ou encore l'identifiant du variant dans dbSNP.

Annexe F

Algorithme de parcours d'un graphe d'assertions

Description

L'algorithme suivant permet l'exploration en profondeur d'un graphe d'assertions $G_a(V, E)$ avec :
- une profondeur maximale p_{max} : la longueur maximale d'un chemin est limitée par p_{max} dont la valeur est précisée par l'analyste en début de processus,
- l'interdiction de faire des boucles : l'algorithme interdit à un même chemin de passer deux fois par le même nœud,
- l'interdiction de revenir en arrière : après avoir emprunté un arc dans un sens, l'algorithme interdit d'emprunter lors de l'étape suivante un arc avec le même label en sens inverse.

Cet algorithme correspond à une *exploration itérative en profondeur* limitée notamment par les trois critères énumérés ci-dessus. Pour plus de lisibilité nous articulons notre algorithme en trois fonctions :
- Exploration-graphe-d-assertion,
- Exploration-profondeur-limitée, et
- EPL-récursive.

La description de ces trois fonctions s'inspire des algorithmes d'*exploration en profondeur limitée* proposées dans le livre de Russel et Norvig [RN03].

Algorithme

Algorithme F.1 Parcours en profondeur d'un graphe d'assertions

```
1 :  fonction Exploration-graphe-d-assertion(G_a, d_max)
       retourne un ensemble de chemins
2 :    ensembleDeChemins := ∅                                   {initialisation}
3 :    pour chaque profondeur ⩽ d_max
4 :      si Exploration-profondeur-limitée ≠ interruption
5 :        ensembleDeChemins := ensembleDeChemins
                                ∪ {Exploration-profondeur-limitée(G_a, profondeur)}
6 :      fin si
7 :    fin pour
8 :    retourner ensembleDeChemins
```

```
9 :  fonction Exploration-profondeur-limitée(G_a, profondeur)
       retourne un ensemble de chemins pour une profondeur donnée ou interruption
10 :   a := Nœud-racine[G_a]                          {a est le nœud racine du graphe G_a}
11 :   V_visité[G_a] := V_visité[G_a] ∪ {a}                {ensemble de nœud visités}
11 :   retourner EPL-récursive(G_a, a, profondeur)   {résultat de la fonction EPL-récursive}
```

```
12 : fonction EPL-récursive(G_a, x, profondeur)
       retourne un chemin ou interruption
13 :   R⁻ := NIL                                          {initialisation d'un arc R⁻}
14 :   pour chaque b ∈ Adjacent[x]                        {pour chaque nœud adjacent à x}
15 :     si b ∉ V_visité[G_a]                          {le nœud adjacent n'a pas été visité}
               et (x, b) ≠ R⁻          {l'arc (x, b) n'est pas l'inverse du dernier arc emprunté}
               et Profondeur[b] ⩽ profondeur          {l'exploration respecte la limite}
16 :       V_visité[G_a] := V_visité[G_a] ∪ {a}
17 :       R⁻ := Inverse[(x, b)]                      {R⁻ est l'arc inverse de celui emprunté}
18 :       chemin := AjouterNœud(chemin, b)
19 :       EPL-récursive(G_a, b, profondeur)
20 :       retourner chemin
21 :     sinon
22 :       retourner interruption
23 :     fin si
24 :   fin pour
```

Annexe G

Captures d'écrans du plugin de Protégé 4 pour l'AAR

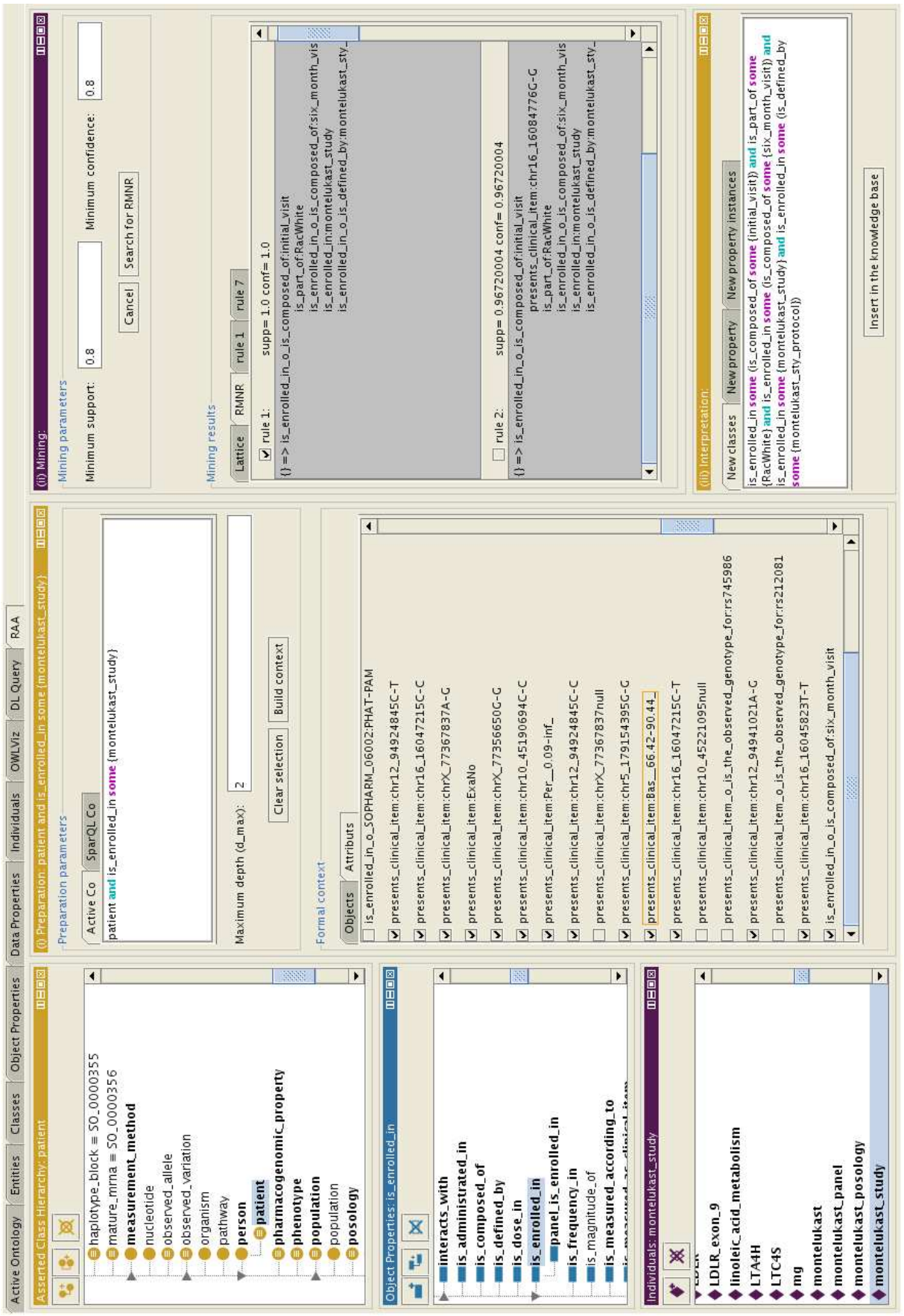

FIG. G.1 – Capture d'écran du plugin de Protégé 4 pour l'Analyse d'Assertions de Rôles

Annexe H

$\mathcal{RMNR}$ extraites de la BC relative à l'investigation clinique du montelukast

Règles associant des génotypes

Règle H.1 *(supp=0,49 ; conf=1)*

{presents_clinical_item :chr16_16045823T-T}	*=>*	*{presents_clinical_item :chr16_16047215T-T}*

Règle H.2 *(supp=0,38 ; conf=96)*

{presents_clinical_item :chr16_16047215C-T}	*=>*	*{presents_clinical_item :chr16_16045823C-T}*

Règle H.3 *(supp=0,64 ; conf=0,95)*

{presents_clinical_item :chr16_15994334C-C}	*=>*	*{presents_clinical_item :chr16_15986618G-G}*

Règle H.4 *(supp=0,44 ; conf=0,90)*

{presents_clinical_item :chr10_45190694C-T, *presents_clinical_item :chr10_45211490A-G}*	*=>*	*{presents_clinical_item :chr10_45221095A-G}*

Règle H.5 *(supp=0,41 ; conf=1)*

{presents_clinical_item :chr10_45198914A-A, *presents_clinical_item :chr10_45221095A-G}*	*=>*	*{presents_clinical_item :chr10_45211490A-G}*

Règle H.6 *(supp=0,39 ; conf=0,86)*

{presents_clinical_item :chrX_77367837A-A}	*=>*	*{presents_clinical_item :chrX_77346486T-T,* *presents_clinical_item :chrX_77389891G-G,* *presents_clinical_item :chrX_77356650G-G,* *presents_clinical_item :chrX_77334462A-A}*

Règle H.7 *(supp=0,31 ; conf=0,95)*

{presents_clinical_item :chrX_77389891A-G, *presents_clinical_item :chrX_77367837A-G}*	*=>*	*{presents_clinical_item :chrX_77334462A-G}*

Règles associant génotype, phénotype, et traitement

Règle H.8 *(supp=0,20 ; conf=0,80)*

{presents_clinical_item :chr12_94941021A-A, *presents_clinical_item :chr16_16143440C-G}*	*=>*	*{presents_clinical_item :ExaNo,* *is_enrolled_in_o_is_defined_by_o_is_composed_of :* *montelukast_treatment}*

Règle H.9 *(supp=0,21 ; conf=0,81)*

{presents_clinical_item :chr10_45211490A-A}	*=>*	*{presents_clinical_item :Per__-inf-0.08_, is_enrolled_in_o_is_defined_by_o_is_composed_of : montelukast_treatment}*

Règle H.10 *(supp=0,26 ; conf=0,80)*

{presents_clinical_item :chrX_77334462A-G}	*=>*	*{presents_clinical_item :chrX_77367837A-G, presents_clinical_item :Per__-inf-0.08_, is_enrolled_in_o_is_defined_by_o_is_composed_of :montelukast_treatment}*

Règle H.11 *(supp=0,20 ; conf=1)*

{presents_clinical_item :chr12_94941021A-G, presents_clinical_item :ExaYes}	*=>*	*{presents_clinical_item :chr16_16024772C-C, is_enrolled_in_o_is_defined_by_o_is_composed_of : montelukast_treatment}*

Règle H.12 *(supp=0,26 ; conf=0,75)*

{presents_clinical_item :chr5_179153244A-C, presents_clinical_item :ExaNo}	*=>*	*{presents_clinical_item :chr16_16024772C-C, is_enrolled_in_o_is_defined_by_o_is_composed_of : montelukast_treatment}*

Bibliographie

[AAD+96] S. Agarwal, R. Agrawal, P. Deshpande, A. Gupta, J.F. Naughton, R. Ramakrishnan, and S. Sarawagi. On the computation of multidimensional aggregates. In *VLDB '96 : Proceedings of the 22th International Conference on Very Large Data Bases*, pages 506–521, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc.

[ABB+00] M. Ashburner, C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Eppig, M.A. Harris, D.P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J.C. Matese, J.E. Richardson, M. Ringwald, G.M. Rubin, and G. Sherlock. Gene Ontology : tool for the unification of biology. *Nat. Genet.*, 25(1) :25–29, 2000.

[ABC+99] R.B. Altman, M. Bada, X.J. Chai, M. Whirl Carrillo, R.O. Chen, and N.F. Abernethy. RiboWeb : An ontology-based system for collaborative molecular biology. *IEEE Intelligent Systems*, 14(5) :68–76, 1999.

[ABG+06] M. Ackermann, B. Berendt, Marko Grobelnik, A. Hotho, D. Mladenic, G. Semeraro, M. Spiliopoulou, G. Stumme, V. Svátek, and M. van Someren, editors. *Semantics, Web and Mining, Joint International Workshops, EWMF 2005 and KDO 2005, Porto, Portugal, October 3 and 7, 2005, Revised Selected Papers*, volume 4289 of *Lecture Notes in Computer Science*. Springer, 2006.

[ABH95] S.S. Anand, D.A. Bell, and J.G. Hughes. The role of domain knowledge in data mining. In *CIKM'95 : Proceedings of the fourth international conference on Information and knowledge management*, pages 37–43, New York, NY, USA, 1995. ACM.

[AEB+08] E. Antezana, M. Egaña, B. De Baets, M. Kuiper, and V. Mironov. ONTO-PERL : An API for supporting the development and analysis of bio-ontologies. *Bioinformatics*, 24(6) :885–887, 2008.

[AFC99] D.B. Aronow, F. Fangfang, and W.B. Croft. Ad hoc classification of radiology reports. *J. Am. Med. Inform. Assoc.*, 6(5) :393–411, 1999.

[AIS93] R. Agrawal, T. Imielinski, and A.N. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, *SIGMOD Conference*, pages 207–216. ACM Press, 1993.

[AK02] R.B. Altman and T. Klein. Challenges for biomedical informatics and pharmacogenomics. *Annu. Rev. Pharmacol. Toxicol.*, 42 :113–33, 2002.

[AMB06] Y. An, J. Mylopoulos, and A. Borgida. Building semantic mappings from databases to ontologies. In *AAAI*. AAAI Press, 2006.

[BA96] R.J. Brachman and T. Anand. The process of knowledge discovery in databases. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 37–57. AAAI/MIT Press, 1996.

[Bar02] M.R. Barnes. SNP and mutation data on the Web - hidden treasures for uncovering. *Comp. Funct. Genomics*, 3(1) :67–74, 2002.

[Bat08] A. Bateman. Editorial. *Nucleic Acids Research*, 36(Database issue D1), 2008.

[Bax06] A.D. Baxevanis. Searching the NCBI databases using Entrez. *Curr. Protoc. Bioinformatics*, 1(3), 2006.

[BB06] C.A. Ball and A. Brazma. Mged standards : work in progress. *Omics*, 10 :138–44, 2006.

[BBL05] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In Kaelbling and Saffiotti [KS05], pages 364–369.

[BC08] BioPAX Consortium. BioPAX : Biological pathways exchange. http ://www.biopax.org/, (dernière consultation : 14 juillet 2008).

[BCBF08] A. Bairoch, S. Cohen-Boulakia, and C. Froidevaux, editors. *Data Integration in the Life Sciences, 5th International Workshop, DILS 2008, Evry, France, June 25-27, 2008. Proceedings*, volume 5109 of *Lecture Notes in Computer Science*. Springer, 2008.

[BCGP04] J. Barrasa, O. Corcho, and A. Gómez-Pérez. R2O, an extensible and semantically based database-to-ontology mapping language. In *Semantic Web and Databases, Second International Workshop, SWDB 2004, Toronto, Canada*, 2004.

[BCM+03] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider, editors. *The Description Logic Handbook : Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[BCM05] P. Buitelaar, P. Cimiano, and B. Magnini. *Ontology Learning from Text : Methods, Evaluation and Applications*, volume 123 of *Frontiers in Artificial Intelligence*. IOS Press, 2005.

[BDdG94] P. Benlian, F. Dairou, and J.L. de Gennes. Apports de la génétique moléculaire dans les hypercholestérolémies pures primitives. *Bulletin de l'Académie nationale de médecine*, 178(3) :393–404, Mar 1994.

[BdLM08] F. Badra, M. d'Aquin, J. Lieber, and T. Meilender. EdHibou : a customizable interface for decision support in a semantic portal. In *International Semantic Web Conference, poster*, 2008.

[BFG+04] P. Buitelaar, J. Franke, M. Grobelnik, G. Paass, and V. Svatek, editors. *Proceedings of the Workshop on Knowledge Discovery and Ontologies at ECML/PKDD'04*, Pisa, Italy, September 2004.

[BFMD05] J. C. Barrett, B. Fry, J. Maller, and M. J. Daly. Haploview : analysis and visualization of LD and haplotype maps. *Bioinformatics*, 21(2) :263–5, Jan 2005.

[BGL08] F. Baader, S. Ghilardi, and C. Lutz. LTL over description logic axioms. In Baader et al. [BLM08].

[BGSS07] F. Baader, B. Ganter, B. Sertkaya, and U. Sattler. Completing description logic knowledge bases using formal concept analysis. In M.M. Veloso, editor, *IJCAI*, pages 230–235, 2007.

[BGvH+03] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt. C-owl : Contextualizing ontologies. In D. Fensel, K.P. Sycara, and J. Mylopoulos, editors, *International Semantic Web Conference*, volume 2870 of *Lecture Notes in Computer Science*, pages 164–179. Springer, 2003.

[BHS02] B. Berendt, A. Hotho, and G. Stumme, editors. *Proceedings of the Workshop on Semantic Web Mining (SWM'02 at ECML/PKDD'02)*, Helsinki, Finland, August 2002.

[BKvH02] J. Broekstra, A. Kampman, and F. van Harmelen. Sesame : A generic architecture for storing and querying RDF and RDF Schema. In I. Horrocks and J.A. Hendler, editors, *International Semantic Web Conference*, volume 2342 of *Lecture Notes in Computer Science*, pages 54–68. Springer, 2002.

[BLHL01] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic Web. *Scientific American*, 284(5) :35–43, May 2001.

[BLM08] F. Baader, C. Lutz, and B. Motik, editors. *Proceedings of the 21st International Workshop on Description Logics (DL2008), Dresden, Germany, May 13-16, 2008*, volume 353 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[BPH05] A. Bernstein, F.J. Provost, and S. Hill. Toward intelligent assistance for a data mining process : An ontology-based approach for cost-sensitive classification. *IEEE Trans. Knowl. Data Eng.*, 17(4) :503–518, 2005.

[Bri06] L. Brisson. *Intégration de connaissances expertes dans un processus de fouille de données pour l'extraction d'informations pertinentes*. Thèse en informatique, Université de Nice - Sophia Antipolis, France, Déc 2006.

[BS85] R.J. Brachman and J.G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2) :171–216, 1985.

[BS04] F. Baader and B. Sertkaya. Applying formal concept analysis to description logics. In Eklund [Ekl04], pages 261–286.

[BSc07] B. Berendt, V. Svàtek, and F. Železnỳ, editors. *Proceedings of the Workshop on Prior Conceptual Knowledge in Machine Learning and Knowledge Discovery (PriCKL'07) at ECML/PKDD'07*, Warsaw, Poland, September 2007.

[BTN08] R. Bendaoud, Y. Toussaint, and A. Napoli. PACTOLE : A methodology and a system for semi-automatically enriching an ontology from a collection of texts. In P.W. Eklund and O. Haemmerlé, editors, *ICCS*, volume 5113 of *Lecture Notes in Computer Science*, pages 203–216. Springer, 2008.

[CBBDF07] S. Cohen-Boulakia, O. Biton, S.B. Davidson, and C. Froidevaux. BioGuideSRS : querying multiple sources with a user-centric perspective. *Bioinformatics*, 23(10) :1301–1303, 2007.

[CCH01] P. Clerkin, P. Cunningham, and P. Hayes. Ontology discovery for the semantic Web using hierarchical clustering. In Bettina Berendt Gerd Stumme, Andreas Hotho, editor, *Proceedings of the Workshop on Semantic Web Mining (SWM'01 at ECML/PKDD'01)*, pages 27–38, Freiburg, Germany, September 2001.

[CCQF05] J. Chabalier, C. Capponi, Y. Quentin, and G. Fichant. ISYMOD : a knowledge warehouse for the identification, assembly and analysis of bacterial integrated systems. *Bioinformatics*, 21(7) :1246–1256, 2005.

[CFCH01] W.W. Chapman, M. Fizman, B.E. Chapman, and P.J. Haug. A comparison of classification algorithms to automatically identify chest x-ray reports that support pneumonia. *J. Biomed. Inform.*, 34(1) :4–14, 2001.

[CG05] D. Calvanese and G. De Giacomo. Data integration : A logic-based perspective. *AI Magazine*, 26(1) :59–70, 2005.

[CGL+98] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Description logic framework for information integration. In *KR*, pages 2–13, 1998.

[CGL+01] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Data integration in data warehousing. *Int. J. Cooperative Inf. Syst.*, 10(3) :237–271, 2001.

[CGL+06] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In P. Doherty, J. Mylopoulos, and C.A. Welty, editors, *KR*, pages 260–270. AAAI Press, 2006.

[CGL+07] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics : The *l-lite* family. *J. Autom. Reasoning*, 39(3) :385–429, 2007.

[CGLR04] D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Logical foundations of peer-to-peer data integration. In A. Deutsch, editor, *PODS*, pages 241–251. ACM, 2004.

[CGLV01] D. Calvanese, G. De Giacomo, M. Lenzerini, and M.Y. Vardi. View-based query answering and query containment over semistructured data. In G. Ghelli and G. Grahne, editors, *DBPL*, volume 2397 of *Lecture Notes in Computer Science*, pages 40–61. Springer, 2001.

[CHS05] P. Cimiano, A. Hotho, and S. Staab. Learning concept hierarchies from text corpora using formal concept analysis. *Journal of the Artifical Intelligence Research*, 24 :305–339, 2005.

[CHST04] P. Cimiano, A. Hotho, G. Stumme, and J. Tane. Conceptual knowledge processing with formal concept analysis and ontologies. In Eklund [Ekl04], pages 189–207.

[CLC+06] T.A. Clayton, J.C. Lindon, O. Cloarec, H. Antti, C. Charuel, G. Hanton, J.P. Provost, J.L. Le Net, D. Baker, R.J. Walley, J.R. Everett, and J.K. Nicholson. Pharmaco-metabonomic phenotyping and personalized drug treatment. *Nature*, 440(7087) :1073–1077, 2006.

[CMF+07] G. Colombo, D. Merico, G. Frisoni, M. Antoniotti, F. De Paoli, and G. Mauri. An ontological modeling approach to neurovascular disease study : the NEUROWEB case. In *Proceedings of the International Workshop on Network Tools and Applications in Biology (NETTAB'07)*, pages 177–186, Pisa, Italy, 2007.

[Con01] The International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409 :860–921, 2001.

[Con03] The International HapMap Consortium. The International HapMap Project. *Nature*, 426 :789–796, 2003.

[Con05] The International HapMap Consortium. A second generation human haplotype map of over 3.1 million snps. *Nature*, 449(7164) :851–861, 2005.

[CR04] C. Carpineto and G. Romano. *Concept Data Analysis : Theory and Applications*. John Wiley & Sons, Chichester, England, 2004.

[CRS+04] H. Cespivova, J. Rauch, V. Svatek, M. Kejkula, and M. Tomeckova. Roles of medical ontology in association mining CRISP-DM Cycle. In P. Buitelaar, J. Franke, M. Grobelnik, G. Paass, and V. Svatek, editors, *Proceedings of the Workshop on Knowledge Discovery and Ontologies at ECML/PKDD'04*, Pisa, Italy, September 2004.

[CSTB+06] A. Coulet, M. Smaïl-Tabbone, P. Benlian, A. Napoli, and M.D. Devignes. SNP-Converter : An ontology-based solution to reconcile heterogeneous SNP descriptions for pharmacogenomic studies. In U. Leser, F. Naumann, and B.A. Eckman, editors, *DILS*, volume 4075 of *Lecture Notes in Computer Science*, pages 82–93. Springer, 2006.

[CSTB+08] A. Coulet, M. Smaïl-Tabbone, P. Benlian, A. Napoli, and M.D. Devignes. Ontology-guided data preparation for discovering genotype-phenotype relationships. *BMC Bioinformatics*, 9(Suppl 4) :S3, 2008.

[CSTND06] A. Coulet, M. Smaïl-Tabbone, A. Napoli, and M.D. Devignes. Suggested Ontology for Pharmacogenomics (SO-Pharm) : Modular construction and preliminary testing. In *Proceedings of the International Workshop on Knowledge Systems in Bioinformatics - KSinBIT'06*, volume LNCS 4277, pages 648–57, 2006.

[CSTND08a] A. Coulet, M. Smaïl-Tabbone, A. Napoli, and M.D. Devignes. Ontology refinement through Role Assertion Analysis : Example in pharmacogenomics. In Baader et al. [BLM08].

[CSTND08b] A. Coulet, M. Smaïl-Tabbone, A. Napoli, and M.D. Devignes. Role Assertion Analysis : a proposed method for ontology refinement through assertion learning. In *Proceedings of the Fourth Starting AI Researchers' Symposium (STAIRS 2008)*, pages 47–58. IOS Press, 2008.

[CWT06] T.H. Cheng, C.P. Wei, and V.S. Tseng. Feature selection for medical data mining : Comparisons of expert judgment and automatic approaches. In *CBMS*, pages 165–170. IEEE Computer Society, 2006.

[dBL+07] M. d'Aquin, F. Badra, S. Lafrogne, J. Lieber, A. Napoli, and L. Szathmary. Case base mining for adaptation knowledge acquisition. In *Proc. of the 20th Intl. Joint Conf. on Artificial Intelligence (IJCAI'07)*, pages 750–755, Hyderabad, India, Jan 2007. Morgan Kaufmann, Inc.

[DCGR98] R. Dieng, O. Corby, A. Giboin, and M. Ribière. Methods and tools for corporate knowledge management. Technical Report RR-3485, INRIA, 1998.

[dDA00] J. den Dunnen and S Antonarakis. Mutation nomenclature extensions and suggestions to describe complex mutations : a discussion. *Hum. Mutat.*, 15(1) :7–12, 2000.

[dDP03] J. den Dunnen and M. Paalman. Standardizing mutation nomenclature : why bother ? *Hum. Mutat.*, 22(3) :181–182, 2003.

[Dev99] K.J. Devlin. *Infosense : Turning Information into Knowledge*. W. H. Freeman & Co., New York, NY, USA, 1999.

[DGDM91] J. Desmeules, M.P. Gascon, P. Dayer, and M. Magistris. Impact of environmental and genetic factors on codeine analgesia. *Eur J Clin Pharmacol.*, 41(1) :23–6, 1991.

[dLN07] M. d'Aquin, J. Lieber, and A. Napoli. La représentation de points de vue dans le système d'aide à la décision en cancérologie KASIMIR. In *Special issue : Vues, Points de vue, rôles et paradigmes proches. Du concept à son exploitation*, volume 13, pages 143–175. Hermes - Lavoisier, 2007.

[DMS05] F. Dau, M.L. Mugnier, and G. Stumme, editors. *Conceptual Structures : Common Semantics for Sharing Knowledge : 13th International Conference on Conceptual Structures, ICCS 2005, Kassel, Germany, July 18-22, 2005. Proceedings*, volume 3596 of *Lecture Notes in Computer Science*. Springer-Verlag Berlin Heidelberg, 2005.

[DPW08] M. Dibernardo, R. Pottinger, and M. Wilkinson. Semi-automatic Web service composition for the life sciences using the BioMoby semantic Web framework. *J. Biomed. Inform.*, March 2008.

[DQ86] R. Ducournau and J. Quinqueton. YAFOOL : encore un langage à objets à base de *frames*. Technical Report 72, INRIA, 1986.

[DS08] K. Dellschaft and S. Staab. *Strategies for the Evaluation of Ontology Learning*. IOS Press, 2008.

[Duc00] R. Ducournau. Des langages à objets aux logiques terminologiques : les systèmes classificatoires. In *Rapport de Recherche 96–030, LIRMM, Montpellier*, 2000.

[EA93] T. Etzold and P. Argos. SRS - an indexing and retrieval tool for flat file data libraries. *Computer Applications in the Biosciences*, 9(1) :49–57, 1993.

[ea01] J. Craig Venter *et al.*. The sequence of the human genome. *Science*, 291 :1304–1351, 2001.

[Ekl04] P.W. Eklund, editor. *Concept Lattices, Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, February 23-26, 2004, Proceedings*, volume 2961 of *Lecture Notes in Computer Science*. Springer, 2004.

[ER95] J. Euzenat and F. Rechenmann. SHIRKA, 10 ans, c'est TROPES ? In A. Napoli, editor, *LMO*, pages 13–34. INRIA, 1995.

[ER99] W.E. Evans and M.V. Relling. Pharmacogenomics : translating functional genomics into rational therapeutics. *Science*, 286(5439) :487–91, 1999.

[ER04] W.E. Evans and M.V. Relling. Moving towards individualized medicine with pharmacogenomics. *Nature*, 429 :464–468, 2004.

[ES07] J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer, 2007.

[FdE08] N. Fanizzi, C. d'Amato, and F. Esposito. Conceptual clustering and its application to concept drift and novelty detection. In S. Bechhofer, M. Hauswirth, J. Hoffmann, and M. Koubarakis, editors, *ESWC*, volume 5021 of *Lecture Notes in Computer Science*, pages 318–332. Springer, 2008.

[FFR97] A. Farquhar, R. Fikes, and J. Rice. The Ontolingua Server : a tool for collaborative ontology construction. *Int. J. Hum.-Comput. Stud.*, 46(6) :707–727, 1997.

[FGPJ97] M. Fernandez, A. Gomez-Perez, and N. Juristo. METHONTOLOGY : from ontological art towards ontological engineering. In *Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering*, pages 33–40, Stanford, USA, 1997.

[Fis87] D.H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2(2) :139–172, 1987.

[Flo05] C.S. Flordellis. The emergence of a new paradigm of pharmacogenomics. *Pharmacogenomics*, 6(5) :515–526, 2005.

[Fou08] The OBO Foundry. OBO Foundry policy document. Technical report, 2008.

[FPSM91] W.J. Frawley, G. Piatetsky-Shapiro, and C.J. Matheus. Knowledge discovery in databases : An overview. In *Knowledge Discovery in Databases*, pages 1–30. AAAI/MIT Press, 1991.

[FPSS96] U.M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery : An overview. In Fayyad et al. [FPSSU96], pages 1–34.

[FPSSU96] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996.

[Fre98] A.A. Freitas. On objective measures of rule surprisingness. In Zytkow and Quafafou [ZQ98], pages 1–9.

[FRS05] S. Ferré, O. Ridoux, and B. Sigonneau. Arbitrary relations in formal concept analysis and logical information systems. In Dau et al. [DMS05], pages 166–180.

[Gai89] B.R. Gaines. An ounce of knowledge is worth a ton of data : quantitative studies of the trade-off between expertise and data based on statistically well-founded empirical induction. In *Proceedings of the sixth international workshop on Machine learning*, pages 156–159, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

[Gan84] B. Ganter. Two basic algorithms in concept analysis. Technical report, Technische Hochschule, Darmstadt, 1984.

[Gan05] A. Gangemi. Ontology design patterns for semantic Web content. In Y. Gil, E. Motta, V. Richard Benjamins, and M.A. Musen, editors, *International Semantic Web Conference*, volume 3729 of *Lecture Notes in Computer Science*, pages 262–276. Springer, 2005.

[GBe07] K.M. Giacomini, C.M. Brett, and R.B. Altman *et al.* . The pharmacogenetics research network from SNP discovery to clinical drug response. *Clinical pharmacology and therapeutics*, 81(3) :328–45, 2007.

[GCCL06] A. Gangemi, C. Catenacci, M. Ciaramita, and J. Lehmann. Modelling ontology evaluation and validation. In Y. Sure and J. Domingue, editors, *ESWC*, volume 4011 of *Lecture Notes in Computer Science*, pages 140–154. Springer, 2006.

[GDF+04] Y. Gasche, Y. Daali, M. Fathi, A. Chiappe, S. Cottini, P. Dayer, and J. Desmeules. Codeine intoxication associated with ultrarapid cyp2d6 metabolism. *N. Engl. J. Med.*, 351(27) :2827–31, 2004.

[GE03] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3 :1157–1182, 2003.

[GKM04] P. Gottgtroy, N. Kasabov, and S. MacDonell. An ontology driven approach for knowledge discovery in biomedicine. In *Proceedings of the 8th Pacific Rim International Conference on Artificial Intelligence*, volume LNAI 3157, pages 53–67, Berlin, Germany, 2004. Springer.

[GMB+05] E. Guérin, G. Marquet, A. Burgun, O. Loréal, L. Berti-Equille, U. Leser, and F. Moussouni. Integrating and warehousing liver gene expression data and related biomedical resources in gedaw. In B. Ludäscher and L. Raschid, editors, *DILS*, volume 3615 of *Lecture Notes in Computer Science*, pages 158–174. Springer, 2005.

[GPCGFL03] A. Gomez-Perez, O. Corcho-Garcia, and M. Fernandez-Lopez. *Ontological Engineering*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.

[Gru93] T.R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2) :199–220, 1993.

[GS08] C. Goble and R. Stevens. State of the nation in data integration for bioinformatics. *J Biomed Inform*, February 2008.

[GSC+08] RA. George, TD. Smith, S. Callaghan, L. Hardman, C. Pierides, O. Horaitis, MA. Wouters, and Cotton RG. General mutation databases : analysis and review. *Journal of Medical Genetics*, 45(2) :65–70, 2008.

[GSK+88] F.J. Gonzalez, R.C. Skoda, S. Kimura, M. Umeno, U.M. Zanger, D.W. Nebert, H.V. Gelboin, J.P. Hardwick, and U.A. Meyer. Characterization of the common genetic defect in humans deficient in debrisoquine metabolism. *Nature*, 331(6155) :442–446, 1988.

[GVM93] A. Guénoche and I. Van Mechelen. Galois approach to the induction of concepts. In *Categories and concepts : Theorical views and inductive data analysis*, pages 287–308. Academic Press, 1993.

[GW99] B. Ganter and R. Wille. *Formal Concept Analysis : Mathematical Foundations*. Springer, 1999.

[GW04] C. Goble and C. Wroe. The Montagues and the Capulets : Conference papers. *Comp. Funct. Genomics*, 5(8) :623–632, 2004.

[Hac04] M.S. Hacid. Special issue on Web data integration, Introduction. *Information Systems*, 29(1) :1–2, 2004.

[Hal01] A.Y. Halevy. Answering queries using views : A survey. *VLDB J.*, 10(4) :270–294, 2001.

[Hal05] A.Y. Halevy. Why your data won't mix. *ACM Queue*, 3(8) :50–58, 2005.

[HBWCH+08] T. Hernandez-Boussard, M. Whirl-Carrillo, J.M. Hebert, L. Gong, R. Owen, M. Gong, W. Gor, F. Liu, C. Truong, R. Whaley, M. Woon, T. Zhou, R.B. Altman, and T.E. Klein. The pharmacogenetics and pharmacogenomics knowledge base : accentuating the knowledge. *Nucleic Acids Res.*, 36(Database issue) :D913–D918, 2008.

[HF94] J. Han and Y. Fu. Dynamic generation and refinement of concept hierarchies for knowledge discovery in databases. In *Proceedings of AAAI'94 Workshop on Knowledge Discovery in Databases (KDD'94)*, pages 157–168, 1994.

[HHNV07] M. Rouane Hacene, M. Huchard, A. Napoli, and P. Valtchev. A proposal for combining formal concept analysis and description logics for mining relational data. In S.O. Kuznetsov and S. Schmidt, editors, *ICFCA*, volume 4390 of *Lecture Notes in Computer Science*, pages 51–65. Springer, 2007.

[H.J02] H.J.Motulsky. *Biostatistique, une approche intuitive*. De Boeck Université, 2002.

[HK01] J. Han and M. Kamber. *Data Mining : Concepts and Techniques*. Morgan Kaufmann, 2001.

[HLTB04] I. Horrocks, L. Li, D. Turi, and S. Bechhofer. The Instance Store : DL reasoning with large numbers of individuals. In V. Haarslev and R. Möller, editors, *Description Logics*, volume 104 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.

[HM03] V. Haarslev and R. Möller. Racer : A core inference engine for the semantic Web. In Y. Sure and O. Corcho, editors, *EON*, volume 87 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.

[HMS05] U. Hustadt, B. Motik, and U. Sattler. Data complexity of reasoning in very expressive description logics. In Kaelbling and Saffiotti [KS05], pages 466–471.

[Hor07] I. Horrocks. OBO flat file format syntax and semantics and mapping to OWL Web ontology language. Technical report, University of Manchester, 2007.

[HPSvH03] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL : the making of a web ontology language. *J. Web Sem.*, 1(1) :7–26, 2003.

[HRT+05] M.L. Hastings, N. Rest, D. Traum, A. Stella, G. Guanti, and A.R. Krainer. An LKBI AT-AC intron mutation causes Peutz-Jeghers syndrome via splicing at noncanonical cryptic splice site. *Struct. Mol. Biol.*, 12(1) :54–59, 2005.

[HSS06] B.M. Hemminger, B. Saelim, and P.F. Sullivan. TAMAL : an integrated approach to choosing SNPs for genetic studies of human complex traits. *Bioinformatics*, 22(5) :626–627, 2006.

[HVK+02] M.K. Higashi, D.L. Veenstra, L.M. Kondo, A.K. Wittkowsky, S.L. Srinouanprachanh, F.M. Farin, and A.E. Rettie. Association between CYP2C9 genetic variants and anticoagulation-related outcomes during warfarin therapy. *JAMA*, 287(13) :1690–1698, 2002.

[HY90] R. Hull and M. Yoshikawa. Ilog : Declarative creation and manipulation of object identifiers. In D. McLeod, R. Sacks-Davis, and H.J. Schek, editors, *VLDB*, pages 455–468. Morgan Kaufmann, 1990.

[JKN08] N. Jay, F. Kohler, and A. Napoli. Analysis of social communities with iceberg and stability-based concept lattices. In Medina and Obiedkov [MO08], pages 258–272.

[KAB06] L. Karoui, M.A. Aufaure, and N. Bennacer. Context-based hierarchical clustering for the ontology learning. In *Web Intelligence*, pages 420–427. IEEE Computer Society, 2006.

[KACV+04] P.D. Karp, M. Arnaud, J. Collado-Vides, J. Ingraham, I.T. Paulsen, and M.H. Jr. Saier. The E. coli EcoCyc database : No longer just a metabolic pathway database. *ASM News*, 70(1) :25–30, 2004.

[Kay97] D. Kayser. *La représentation des connaissances*. collection informatique, hermès edition, 1997.

[KCH+02] P. Kogut, S. Cranefield, L. Hart, M. Dutra, K. Baclawski, M. Kokar, and J. Smith. UML for ontology development. *Knowl. Eng. Rev.*, 17(1) :61–64, 2002.

[KDK+05] R. Karchin, M. Diekhans, L. Kelly, D.J. Thomas, U. Pieper, N. Eswar, D. Haussler, and A. Sali. LS-SNP : large-scale annotation of coding non-synonymous SNPs based on multiple information sources. *Bioinformatics*, 21(12) :2814–2820, 2005.

[KFNM04] H. Knublauch, R.W. Fergerson, N. Fridman Noy, and M.A. Musen. The Protégé OWL plugin : An open development environment for semantic Web applications. In S.A. McIlraith, D. Plexousakis, and F. van Harmelen, editors, *International Semantic Web Conference*, volume 3298 of *Lecture Notes in Computer Science*, pages 229–243. Springer, 2004.

[KJ97] R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artif. Intell.*, 97(1–2) :273–324, 1997.

[KK07] F. Karel and J. Klema. Quantitative association rule mining in genomics using apriori knowledge. In *Workshop on Prior Conceptual Knowledge in Machine Learning and Knowledge Discovery (PriCKL'07) at ECML/PKDD 2007*, pages 53–64, Warsaw, Poland, September 2007.

[KKS+04] A. Kasprzyk, D. Keefe, D. Smedley, D. London, W. Spooner, C. Melsopp, M. Hammond, P. Rocca-Serra, T. Cox, and E. Birney. Ensmart : A generic system for fast and flexible access to biological data. *Genome Res.*, 14(1) :160–169, 2004.

[KLW08] P.D. Karp, T.J. Lee, and V. Wagner. BioWarehouse : Relational integration of eleven bioinformatics databases and formats. In Bairoch et al. [BCBF08], pages 5–7.

[KLWW08] B. Konev, C. Lutz, D. Walther, and F. Wolter. Semantic modularity and module extraction in description logics. In *ECAI 2008, 18th European Conference on Artificial Intelligence, Patras, Greece, Proceedings*, pages 55–59, 2008.

[KN01] L. Kruglyak and D.A. Nickerson. Variation is the spice of life. *Nature Genetics*, 27 :234–236, 2001.

[KO02] S.O. Kuznetsov and S.A. Obiedkov. Comparing performance of algorithms for generating concept lattices. *J. Exp. Theor. Artif. Intell.*, 14(2–3) :189–216, 2002.

[KPL03] J. Köhler, S. Philippi, and M. Lange. SEMEDA : ontology based semantic integration of biological databases. *Bioinformatics*, 19(18) :2420–2427, 2003.

[KPS+06] A. Kalyanpur, B. Parsia, E. Sirin, B. Cuenca Grau, and J.A. Hendler. Swoop : A Web ontology editing browser. *J. Web Sem.*, 4(2) :144–153, 2006.

[Kry02] M. Kryszkiewicz. Concise representations of association rules. In *Proceedings of the ESF Exploratory Workshop on Pattern Detection and Discovery*, pages 92–109, London, UK, 2002. Springer-Verlag.

[KS05] L. Pack Kaelbling and A. Saffiotti, editors. *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30-August 5, 2005*. Professional Book Center, 2005.

[Kuz07] S.O. Kuznetsov. On stability of a formal concept. *Ann. Math. Artif. Intell.*, 49(1–4) :101–115, 2007.

[Len02] M. Lenzerini. Data integration : A theoretical perspective. In L. Popa, editor, *PODS*, pages 233–246. ACM, 2002.

[LFZ99] N. Lavrac, P.A. Flach, and B. Zupan. Rule evaluation measures : A unifying view. In S. Dzeroski and P.A. Flach, editors, *ILP*, volume 1634 of *Lecture Notes in Computer Science*, pages 174–185. Springer, 1999.

[LHCM00] B. Liu, W. Hsu, S. Chen, and Y. Ma. Analyzing the subjective interestingness of association rules. *IEEE Intelligent Systems*, 15(5) :47–55, 2000.

[LN05] U. Leser and F. Naumann. (almost) hands-off information integration for the life sciences. In *CIDR*, pages 131–143, 2005.

[LNST08] J. Lieber, A. Napoli, L. Szathmary, and Y. Toussaint. First elements on Knowledge Discovery guided by Domain Knowledge (kddk). In S. B. Yahia, E. M. Nguifo, and R. Belohlavek, editors, *Concept Lattices and Their Applications (CLA 06)*, Lecture Notes in Artificial Intelligence 4923, pages 22–41. Springer, Berlin, 2008.

[LWZ08] C. Lutz, F. Wolter, and M. Zakharyaschev. Temporal description logics : A survey. In *Proceedings of the 15th International Symposium on Temporal Representation and Reasoning, time*, pages 3–14, 2008.

[LY05] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans. Knowl. Data Eng.*, 17(4) :491–502, 2005.

[LZG+06] J.J. Lima, S. Zhang, A. Grant, L. Shao, K.G. Tantisira, H. Allayee, J. Wang, J. Sylvester, J. Holbrook, R. Wise, S.T. Weiss, and K. Barnes. Influence of leukotriene pathway polymorphisms on response to montelukast in asthma. *Am. J. Respir. Crit. Care Med.*, 173(4) :379–85, 2006.

[Mar03] F.M. Marincola. Translational medicine : A two-way road. *Journal of Translational Medicine*, 1(1) :1, 2003.

[Mau06] S. Maumus. *Approche de la complexité du syndrome métabolique et de ses indicateurs de risque par la mise en oeuvre de méthodes numériques et symboliques de fouille de données.* Thèse en épidémiologie et santé publique, Université Henri Poincaré – Nancy 1, France, Nov 2006.

[MBB+08] F. Mougin, A. Burgun, O. Bodenreider, J. Chabalier, O. Loréal, and P. Le Beux. Automatic methods for integrating biomedical data sources in a mediator-based system. In Bairoch et al. [BCBF08], pages 61–76.

[McG05] Ken McGarry. A survey of interestingness measures for knowledge discovery. *The Knowledge Engineering Review*, 20(1) :39–61, 2005.

[McK98] V.A. McKusick. Mendelian inheritance in man. a catalog of human genes and genetic disorders, 1998.

[MD07a] D. Martin and J. Domingue. Semantic Web services : Part 1. *IEEE Intelligent Systems*, 22(5) :12–17, 2007.

[MD07b] D. Martin and J. Domingue. Semantic Web services : Part 2. *IEEE Intelligent Systems*, 22(6) :8–15, 2007.

[MDNST05] N. Messai, M.D. Devignes, A. Napoli, and M. Smaïl-Tabbone. Querying a bioinformatic data sources registry with concept lattices. In Dau et al. [DMS05], pages 323–336.

[MFJ+07] M.D. Mailman, M. Feolo, Y. Jin, M. Kimura, K. Tryka, R. Bagoutdinov, L. Hao, A. Kiang, J. Paschall, L. Phan, N. Popova, S. Pretel, L. Ziyabari, M. Lee, Y. Shao, Z.Y. Wang, K. Sirotkin, M. Ward, M. Kholodov, K. Zbicz, J. Beck, M. Kimelman, S. Shevelev, D. Preuss, E. Yaschenko, A. Graeff, J. Ostell, and S.T. Sherry. The NCBI dbGaP database of genotypes and phenotypes. *Nat. Genet.*, 39(10) :1181–1186, 2007.

[MK08] P. Hitzler M. Krötzsch, S. Rudolph. Description logic rules. In *ECAI 2008, 18th European Conference on Artificial Intelligence, 2008, Patras, Greece, Proceedings*, pages 80–84, 2008.

[MKS04] H.M. Müller, E.E. Kenny, and P.W. Sternberg. Textpresso : an ontology-based information retrieval and extraction system for biological literature. *PLoS Biol*, 2(11) :e309, 2004.

[MMK+03] R.H. Mathijssen, S. Marsh, M.O. Karlsson, R. Xie, S.D. Baker, J. Verweij, A. Sparreboom, and H.L. McLeod. Irinotecan pathway genotype analysis to predict pharmacokinetics. *Clin. Cancer Res.*, 9(9) :3246–3253, 2003.

[MO08] R. Medina and S.A. Obiedkov, editors. *Formal Concept Analysis, 6th International Conference, ICFCA 2008, Montreal, Canada, February 25-28, 2008, Proceedings*, volume 4933 of *Lecture Notes in Computer Science*. Springer, 2008.

[MTB+99] I. Martinelli, E. Taioli, P. Bucciarelli, S. Akhavan, and P.M. Mannucci. Interaction between the G20210A mutation of the prothrombin gene and oral contraceptive use in deep vein thrombosis. *Arterioscler. Thromb. Vasc. Biol.*, 19(3) :700–703, 1999.

[Mug91] Stephen Muggleton. Inductive Logic Programming. *New Generation Comput.*, 8(4) :295, 1991.

[MVB+95] C. Médigue, T. Vermat, G. Bisson, A. Viari, and A. Danchin. Cooperative computer system for genome sequence analysis. In C.J. Rawlings, D.A. Clark, R.B. Altman, L. Hunter, T. Lengauer, and S.J. Wodak, editors, *ISMB*, pages 249–258. AAAI, 1995.

[MZCC04] A.A. Mitchell, M.E. Zwick, A. Chakravarti, and D.J. Cutler. Discrepancies in dbSNP confirmation rates and allele frequency distributions from varying genotyping error rates and patterns. *Bioinformatics*, 20(7) :1022–1032, 2004.

[Nap92] A. Napoli. *Représentations à objets et raisonnement par classification en intelligence artificielle*. Doctorat d'état ès sciences mathématiques, Université Henri Poincaré – Nancy 1, France, Jan 1992.

[NB04] Z. Nazeri and E. Bloedorn. Exploiting available domain knowledge to improve mining aviation safety and network security data. In P. Buitelaar, J. Franke, M. Grobelnik, G. Paass, and V. Svatek, editors, *Proceedings of the Workshop on Knowledge Discovery and Ontologies at ECML/PKDD'04*, Pisa, Italy, September 2004.

[NBS+06] M.C.Y. Ng, L. Baum, W.Y. So, V.K.L. Lam, Y. Wang, E. Poon, B. Tomlinson, S. Cheng, K. Lindpaintner, and J.C.N. Chan. Association of lipoprotein lipase S447X, apolipoprotein E exon 4, and apoC3 -455T-C polymorphisms on the susceptibility to diabetic nephropathy. *Clin Genet*, 70 :20–28, 2006.

[NED00] A. Napoli, J. Euzenat, and R. Ducournau. Les représentations des connaissances par objets. *Technique et Science Informatiques*, 19(1–3) :387–394, 2000.

[NM01] N. F. Noy and D. McGuinness. Ontology development 101 : A guide to creating your first ontology. Technical report, 2001.

[NMG05] A.C. Need, A.G. Motulsky, and D.B. Goldstein. Priorities and standards in pharmacogenetic research. *Nat. Genet.*, 37(7) :671–81, 2005.

[NR06] N. F. Noy and A. Rector. Defining n-ary relations on the semantic Web. Technical report, W3C, 2006.

[Ome01] B. Omelayenko. Learning of ontologies for the Web : the analysis of existent approaches. 2001.

[ORS+02] D.E. Oliver, D.L. Rubin, J.M. Stuart, M. Hewett, T.E. Klein, and R.B. Altman. Ontology development for a pharmacogenetics knowledge base. In *Pacific Symposium on Biocomputing*, pages 65–76, 2002.

[ORT08] R.P. Owen, Altman R.B., and Klein T.E. PharmGKB and the international warfarin pharmacogenetics consortium : the changing role for pharmacogenomic databases and single-drug pharmacogenetics. *Hum. Mutat.*, 29(4) :456–460, 2008.

[PB01] J. Phillips and B.G. Buchanan. Ontology-guided knowledge discovery in databases. In *K-CAP'01 : Proceedings of the 1st international conference on Knowledge capture*, pages 123–130, New York, NY, USA, 2001. ACM.

[PGC+01] M. Page, J. Gensel, C. Capponi, C. Bruley, P. Genoud, D. Ziébelin, D. Bardou, and V. Dupierris. A new approach in object-based knowledge representation : The AROM system. In L. Monostori, J. Váncza, and M. Ali, editors, *IEA/AIE*, volume 2070 of *Lecture Notes in Computer Science*, pages 113–118. Springer, 2001.

[PLC+08] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *J. Data Semantics*, 10 :133–173, 2008.

[PRAC06] D. Pérez-Rey, A. Anguita, and J. Crespo. Ontodataclean : Ontology-based integration and preprocessing of distributed data. In N. Maglaveras, I. Chouvarda, V. Koutkias, and R.W. Brause, editors, *ISBMDA*, volume 4345 of *Lecture Notes in Computer Science*, pages 262–272. Springer, 2006.

[RAC+06] M.C. Rousset, P. Adjiman, P. Chatalic, F. Goasdoué, and L. Simon. Somewhere in the semantic Web. In J. Wiedermann, G. Tel, J. Pokorný, M. Bieliková, and J. Stuller, editors, *SOFSEM*, volume 3831 of *Lecture Notes in Computer Science*, pages 84–99. Springer, 2006.

[RBJ00] J. Rumbaugh, G. Booch, and I. Jacobson. *Le guide de l'utilisateur UML*. Eyrolles, 2000.

[RDM05] D.L. Rubin, O. Dameron, and M.A. Musen. Use of description logic classification to reason about consequences of penetrating injuries. In *Proceedings of the AMIA Annu. Symp.*, pages 649–653, 2005.

[Rec00] F. Rechenmann. From data to knowledge. *Bioinformatics*, 16(5) :411, 2000.

[RFG+02] M.C. Rousset, C. Froidevaux, H. Gagliardi, F. Goasdoué, C. Reynaud, and B. Safar. Construction de médiateurs pour intégrer des sources d'information multiples et hétérogènes : le projet PICSEL. *Revue I3*, 2(1), 2002.

[RIF+06] R. Redon, S. Ishikawa, K.R. Fitch, L. Feuk, G.H. Perry, and *et al.* Global variation in copy number in the human genome. *Nature*, 444 :444–454, 2006.

[RKC06] H.Z. Ring, P.Y. Kwok, and R.G. Cotton. Human variome project : an international collaboration to catalogue human genetic variation. *Pharmacogenomics*, 7(7) :969–972, 2006.

[RKK+00] A. Rzhetsky, T. Koike, S. Kalachikov, S.M. Gomez, M. Krauthammer, S.H. Kaplan, P. Kra, J.J. Russo, and C. Friedman. A knowledge model for analysis and simulation of regulatory networks. *Bioinformatics*, 16(11) :1120–1128, 2000.

[RKM+05] C. Rosse, A. Kumar, J.L. Mejino, D.L. Cook, L.T. Detwiler, and B. Smith. A strategy for improving and integrating biomedical ontologies. In *Proceedings of the AMIA Annu. Symp.*, pages 639–643, 2005.

[RLSN08] M.J. Rieder, R.J. Livingston, I.B. Stanaway, and D.A. Nickerson. The environmental genome project : reference polymorphisms for drug metabolism genes and genome-wide association studies. *Drug Metabolism Review*, 40(2) :241–261, 2008.

[RMKM08] D.L. Rubin, D.A. Moreira, P.P. Kanjamala, and Musen M.A. BioPortal : A Web portal to biomedical ontologies. *2008 AAAI Spring Symposium Series, Symbiotic Relationships between Semantic Web and Knowledge Engineering*, 2008.

[RMM+98] C. Rosse, J.L. Mejino, B.R. Modayur, R. Jakobovits, K.P. Hinshaw, and J.F. Brinkley. Motivation and organizational principles for anatomical knowledge representation : the digital anatomist symbolic knowledge base. *J. Am. Med. Inform.x Assoc.*, 5(1), 1998.

[RN03] S. Russell and P. Norvig. *Artificial Intelligence - A modern approch*. Englewood Cliffs, NJ : Prentice-Hall (2d Edition), 2003.

[RSN07] D.L. Rubin, N.H. Shah, and N.F. Noy. Biomedical ontologies : a functional perspective. *Briefings in Bioinformatics*, 9(1) :75–90, 2007.

[Rud06] S. Rudolph. *Relational Exploration : Combining Description Logics and Formal Concept Analysis for Knowledge Specification*. Thèse en informatique, Technischen Universität – Dresden, Germany, Dec 2006.

[SA95] R. Srikant and R. Agrawal. Mining generalized association rules. In U. Dayal, P.M.D. Gray, and S. Nishio, editors, *VLDB*, pages 407–419. Morgan Kaufmann, 1995.

[SAA+99] G. Schreiber, H. Akkermans, A. Anjewierden, R. Dehoog, N. Shadbolt, W. Vandevelde, and B. Wielinga. *Knowledge Engineering and Management : The CommonKADS Methodology*. The MIT Press, December 1999.

[Sah02] S. Sahar. Exploring interestingness through clustering : A framework. In *ICDM*, pages 677–680. IEEE Computer Society, 2002.

[SAR+07] B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters, L.J. Goldberg, K. Eilbeck, A. Ireland, C.J. Mungall, The OBI Consortium, N. Leontis, P. Rocca-Serra, A. Ruttenberg, S.A. Sansone, R.H. Scheuermann, N. Shah, P.L. Whetzel, and S. Lewis. The OBO Foundry : coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology*, 25(11) :1251–1255, 2007.

[Saï07] F. Saïs. *Intégration sémantique de données guidée par une ontologie*. Thèse en informatique, Université Paris-Sud, France, Déc 2007.

[SBB+00] R. Stevens, P.G. Baker, S. Bechhofer, G. Ng, A. Jacoby, N.W. Paton, C.A. Goble, and A. Brass. Tambis : Transparent access to multiple bioinformatics information sources. *Bioinformatics*, 16(2) :184–186, 2000.

[Ser07] B. Sertkaya. *Formal Concept Analysis Methods for Description Logics*. Thèse en informatique, Technischen Universität – Dresden, Germany, Nov 2007.

[SHB01] G. Stumme, A. Hotho, and B. Berendt, editors. *Proceedings of the Workshop on Semantic Web Mining (SWM'01 at ECML/PKDD'01)*, Freiburg, Germany, September 2001.

[SHSD08] B. Séguin, B. Hardy, P.A. Singer, and A.S. Daar. Bidil : recontextualizing the race debate. *The Pharmacogenomics Journal*, 8 :169–173, 2008.

[SIL05] Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19) :2507–2517, 2005.

[SNK07] L. Szathmary, A. Napoli, and S. O. Kuznetsov. ZART : A Multifunctional Itemset Mining Algorithm. In *Proc. of the 5th Intl. Conf. on Concept Lattices and Their Applications (CLA'07)*, pages 26–37, Montpellier, France, Oct 2007.

[SNV07] L. Szathmary, A. Napoli, and P. Valtchev. Towards Rare Itemset Mining. In *Proc. of the 19th IEEE Intl. Conf. on Tools with Artificial Intelligence (ICTAI'07)*, volume 1, pages 305–312, Patras, Greece, Oct 2007.

[SP04] E. Sirin and B. Parsia. Pellet : An OWL DL reasoner. In *Proceedings of the International Workshop on Description Logics (DL2004)*, 2004.

[Spe08] G. Spencer. International consortium announces the 1000 genomes project. *EMBARGOED*, 2008.

[SRR05] V. Svátek, J. Rauch, and M. Ralbovský. Ontology-enhanced association mining. In Ackermann et al. [ABG+06], pages 163–179.

[Ste08] L.D. Stein. Towards a cyberinfrastructure for the biological sciences : progress, visions and challenges. *Nature Genetics*, 9(9) :678–688, 2008.

[SWK+01] S.T. Sherry, M.H. Ward, M. Kholodov, J. Baker, L. Phan, E.M. Smigielski, and K. Sirotkin. dbSNP : the NCBI database of genetic variation. *Nucleic Acids Res.*, 29(1) :308–311, 2001.

[SWW98] G. Stumme, R. Wille, and U. Wille. Conceptual knowledge discovery in databases using formal concept analysis methods. In Zytkow and Quafafou [ZQ98], pages 450–458.

[Sza06] L. Szathmary. *Symbolic Data Mining Methods with the Coron Platform*. Thèse en informatique, Université Henri Poincaré – Nancy 1, France, Nov 2006.

[TH06] D. Tsarkov and I. Horrocks. FaCT++ description logic reasoner : System description. In *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR 2006)*, volume 4130 of *Lecture Notes in Artificial Intelligence*, pages 292–297. Springer, 2006.

[TKS02] P.N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *KDD'02 : Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 32–41, New York, NY, USA, 2002. ACM.

[TZY+04] A.L. Taylor, S. Ziesche, C. Yancy, P. Carson, R. Jr D'Agostino, K. Ferdinand, M. Taylor, K. Adams, M. Sabolinski, M. Worcel, J.N. Cohn, and African-American Heart Failure Trial Investigators. Combination of isosorbide dinitrate and hydralazine in blacks with heart failure. *N Engl J Med.*, 351(20) :2049–57, 2004.

[UK95] M. Uschold and M. King. Towards a methodology for building ontologies. In *Proceedings of the IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing*, 1995.

[VB05] M. Vanzin and K. Becker. Ontology-based rummaging mechanisms for the interpretation of Web usage patterns. In Ackermann et al. [ABG+06], pages 180–195.

[vHY04] V. van Heyningen and P.L. Yeyati. Mechanisms of non-mendelian inheritance in genetic disease. *Human Molecular Genetics*, 13(RI2) :R225–R233, 2004.

[VMG04] P. Valtchev, R. Missaoui, and R. Godin. Formal concept analysis for knowledge discovery and data mining : The new challenges. In Eklund [Ekl04], pages 352–371.

[VR08] J. Völker and S. Rudolph. Lexico-logical acquisition of OWL DL axioms. [MO08], pages 62–77.

[Web97] W.W. Weber. *Pharmacogenetics*. Oxford University Press, New York NY, 1997.

[WH03] A.B. Wilcox and G. Hripcsak. The role of domain knowledge in automating medical text report classification. *J. Am. Med. Inform. Assoc.*, 10(4) :330–338, 2003.

[Wil02] R. Wille. Why can concept lattices support knowledge discovery in databases ? *J. Exp. Theor. Artif. Intell.*, 14(2–3) :81–92, 2002.

[WLT+06] K. Wolstencroft, P. Lord, L. Tabernero, A. Brass, and R. Stevens. Protein classification using ontology classification. *Bioinformatics*, 22(14) :e530–e538, 2006.

[WMF+08] S.T. Weiss, H.L. McLeod, D.A. Flockhart, M.E. Dolan, N.L. Benowitz, J.A. Johnson, M.J. Ratain, and K.M. Giacomini. Creating and evaluating genetic tests predictive of drug response. *Nat. Rev. Drug. Discov.*, 7(7) :568–74, 2008.

[WMS+05] K. Wolstencroft, R. McEntire, R. Stevens, L. Tabernero, and A. Brass. Constructing ontology-driven protein family databases. *Bioinformatics*, 21(8) :1685–1692, 2005.

[WSGA03] C. Wroe, R. Stevens, C.A. Goble, and M. Ashburner. A methodology to migrate the gene ontology to a description logic environment using daml+oil. In *Pacific Symposium on Biocomputing*, pages 624–635, 2003.

[WZTS05] J.T.L. Wang, M.J. Zaki, H.T.T. Toivonen, and D.E. (Eds.) Shasha. *Data Mining in Bioinformatics*. Springer, 2005.

[YHTL08] H.H. Yang, N. Hu, P.R. Taylor, and M.P. Lee. Whole genome-wide association study using affymetrix SNP chip : a two-stage sequential selection method to identify genes that increase the risk of developing complex diseases. *Methods Mol Med.*, 141 :23–35, 2008.

[YL04] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5 :1205–1224, 2004.

[YSS07] T. Yu, S.J. Simoff, and D. Stokes. Incorporating prior domain knowledge into a kernel based feature selection algorithm. In Z.H. Zhou, H. Li, and Q. Yang, editors, *PAKDD*, volume 4426 of *Lecture Notes in Computer Science*, pages 1064–1071. Springer, 2007.

[ZQ98] J.M. Zytkow and M. Quafafou, editors. *Principles of Data Mining and Knowledge Discovery, Second European Symposium, PKDD '98, Nantes, France, September 23-26, 1998, Proceedings*, volume 1510 of *Lecture Notes in Computer Science*. Springer, 1998.

Résumé

Cette thèse porte sur l'utilisation d'ontologies et de bases de connaissances pour guider différentes étapes du processus d'extraction de connaissances *à partir* de bases de données (ECBD) et une application dans le domaine de la pharmacogénomique. Les données relatives à ce domaine sont hétérogènes, complexes, et distribuées dans diverses bases de données, ce qui rend cruciale l'étape préliminaire de préparation et d'intégration des données à fouiller. Je propose pour guider cette étape une approche originale d'intégration de données qui s'appuie sur une représentation des connaissances du domaine sous forme de deux ontologies en logiques de description : SNP-Ontology et SO-Pharm. Cette approche a été implémentée grâce aux technologies du Web sémantique et conduit au peuplement d'une base de connaissances pharmacogénomique. Le fait que les données à fouiller soient alors disponibles dans une base de connaissances entraîne de nouvelles potentialités pour le processus d'extraction de connaissances. Je me suis d'abord intéressé au problème de la sélection des données les plus pertinentes à fouiller en montrant comment la base de connaissances peut être exploitée dans ce but. Ensuite j'ai décrit et appliqué à la pharmacogénomique, une méthode qui permet l'extraction de connaissances directement *à partir* d'une base de connaissances. Cette méthode appelée Analyse des Assertions de Rôles (ou AAR) permet d'utiliser des algorithmes de fouille de données sur un ensemble d'assertions de la base de connaissances pharmacogénomique et d'expliciter des connaissances nouvelles et pertinentes qui y étaient enfouies.

Mots-clés: extraction de connaissances à partir de bases de données, intégration de données, sélection de données, représentation des connaissances, ontologie, base de connaissances, logiques de description, SNP, pharmacogénomique.

Abstract

This thesis studies the use of ontology and knowledge base for guiding various steps of the Knowledge Discovery in Databases (KDD) process in the domain of pharmacogenomics. Data related to this domain are heterogeneous, complex, and disseminated through several data sources. Consequently, the preliminary step that consists in the preparation and the integration of data is crucial. For guiding this step, an original approach is proposed, based on a knowledge representation of the domain within two ontologies in description logics : SNP-Ontology and SO-Pharm. This approach has been implemented using semantic Web technologies and leads finally to populating a pharmacogenomic knowledge base. As a result, data to analyze are represented in the knowledge base, which is a benefit for guiding following steps of the knowledge discovery process. Firstly, I study this benefit for feature selection by illustrating how the knowledge base can be used for this purpose. Secondly, I describe and apply to pharmacogenomics a new method named Role Assertion Analysis (or RAA) that enables knowledge discovery directly from knowledge bases. This method uses data mining algorithms over assertions of our pharmacogenomic knowledge base and results in the discovery of new and relevant knowledge.

Keywords: knowledge discovery in databases, data integration, feature selection, knowledge representation, ontology, knowledge base, description logics, SNP, pharmacogenomics.