Treillis de Galois et ontologies de domaine pour la classification et la recherche de sources de données génomiques

Nizar Messai

Université Henri Poincaré – Nancy I

Treillis de Galois et ontologies de domaine pour la classification et la recherche de sources de données génomiques

MÉMOIRE

soutenu le 21 Juin 2004

pour l'obtention du

DEA Informatique de Lorraine - École Doctorale IAEM Lorraine

par

Nizar Messai

Composition du jury

	Noëlle Carbonell	Professeur UHP-Nancy1
	Olivier Festor	Directeur de recherche INRIA
	Didier Galmiche	Professeur UHP-Nancy1
	Dominique Mery	Professeur UHP-Nancy1 et ESIAL
Encadrants :	Marie-Dominique Devignes	Chargée de recherche (CR1)au CNRS
	Amedeo Napoli	Directeur de recherche CNRS
	Malika Smaïl-Tabbone	Maitre de conférences UHP-Nancy1

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

Remerciements

C'est un grand plaisir pour moi autant qu'un devoir de remercier toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce travail.

Je tiens d'abord à remercier vivement M. Amedeo Napoli, responsable du projet ORPAILLEUR, pour l'opportunité qu'il m'a offert en m'accueillant au sein de son équipe ainsi que pour son précieux encadrement.

Mes vifs remerciements vont aussi à Mmes Marie-Dominique Devignes et Malika Smaïl-Tabbone, mes deux autres encadrantes, pour leurs conseils et leur soutien tout au long du stage.

Mes remerciements s'adressent aussi à tous les membres de l'équipe ORPAILLEUR en particulier Mlle Shazia Osman pour m'avoir permis de la solliciter pour ses larges connaissances biologiques.

Je dédie ce travail
à ma mère, à mon père,
à ma sœur, à mes deux frères,
à toute ma grande famille
et à tous mes amis

Résumé

Les banques de données génomiques accessibles via le web sont multiples et hétérogènes. Le manque de documentation et la difficulté d'interaction avec ces banques de données exigent des utilisateurs une double compétence informatique et biologique pour pouvoir tirer profil de leur contenu qui reste encore sous-exploité.

Dans ce mémoire nous présentons une méthode de classification et de recherche de sources génomiques pertinentes pour une question donnée en utilisant les treillis de Galois. Elle consiste à construire le treillis de Galois à partir de la relation binaire entre les sources et leurs propriétés. Ces dernières sont extraites à partir d'un ensemble de métadonnées associées aux sources. Un concept construit à partir d'une requête utilisateur est ensuite inséré dans le treillis. Le calcul du résultat se ramène à extraire l'ensemble des sources figurant dans les extensions des subsumants du concept requête dans le treillis de Galois résultant. L'ordre de pertinence des sources est déduit à partir de l'ordre de spécialisation des concepts correspondants dans le treillis. Une amélioration de la méthode consiste à enrichir la requête à partir d'ontologies de domaine avant de l'insérer dans le treillis. Trois modes d'enrichissement sont possibles : l'enrichissement par généralisation, l'enrichissement par spécialisation et l'enrichissement mixte.

Mots-clés: treillis de Galois, ontologie, recherche d'information, enrichissement sémantique, sources génomiques, classification.

Abstract

Genomic data banks available on the web are multiple and heterogenous. The lack of documentation and the difficulty of interaction with these data banks require users competences in both informatic and biological fields for a best use of sources's contents that still remain under exploited.

In this document we present an approach to classify and search relevant genomic sources for a given question in a Galois lattice. It consists in building the Galois lattice from the binary relation between sources and their properties. These properties are extracted from a set of metadata associated to the sources. The concept built from a given user request is then merged into the Galois lattice. The result is given by the extraction of the set of sources belonging to the extensions of the request concept subsumers in the resulting Galois lattice. The sources ranking is given by the concept specificity order in the Galois lattice. An improvement of the approach consists in expanding the query from ontologies before merging it in the lattice. Three forms of expansion are possible : the generalization expansion, the specialization expansion and mixed expansion.

Keywords: Galois lattices, ontology, information retrieval, semantic enrichment, genomic sources, classification.

Table des matières

Table des figures

Liste des tableaux

Introduction générale

Le travail présenté dans ce mémoire a pour cadre général la fouille du web guidée par la sémantique et appliquée aux ressources génomiques. Il s'est déroulé dans l'équipe ORPAILLEUR du LORIA.

La fouille du web est définie comme étant l'application des techniques de fouille de données (datamining) au contenu, à la structure et à l'usage des ressources du web [BHS02, KB00]. À partir de cette définition on distingue trois catégories de fouille du web à savoir la fouille du contenu, la fouille de la structure et la fouille de l'usage. La fouille du contenu est une sorte de fouille de texte, elle décrit la découverte d'informations utiles à partir du contenu ou des données du web. La fouille de la structure essaie de découvrir le modèle de structure sous-jacent aux liens du web. Enfin la fouille de l'usage exploite les données secondaires dérivées à partir des interactions des utilisateurs avec le web. Nous nous intéressons plus particulièrement à la fouille du contenu des ressources. Le contenu des ressources est composé d'une diversité de données du web telles que les données textuelles, ou les métadonnées,et bien d'autres types. Ces données sont généralement non structurées mais elles peuvent aussi être structurées ou semi-structurées. Il est envisageable d'effectuer la fouille du contenu du web entier mais il serait plus intéressant de la limiter à un ensemble de ressources relatives au domaine à un domaine d'intérêt.

Suite à l'évolution de la recherche dans le domaine biologique, un grand nombre de données est rendu accessible via le web. Ces données sont répertoriées dans des sources génomiques offrant des interfaces d'interrogation et par suite un accès plus facile aux informations qu'elles contiennent. La diversité de ces sources et la complémentarité des données qu'elles contiennent permettent aux utilisateurs d'avoir des informations plus complètes. Cependant, l'absence d'un schéma unique, l'incompatibilité des formats de données et l'absence (ou la faible fréquence) de mise à jour du contenu des sources peuvent entraîner des incohérences au niveau des réponses aux questions (requêtes) des utilisateurs. Face à un tel problème, il est indispensable d'avoir une classification des sources selon des informations supplémentaires permettant à l'utilisateurs de juger la pertinence des sources et d'effectuer le choix de l'information correspondant à sa questions.

Problématique

L'enjeu majeur de ce travail est de fournir aux utilisateurs les sources les plus pertinentes susceptibles de répondre à leurs besoins. Il s'agit donc de classer l'ensemble des sources de façon à pouvoir effectuer un choix selon la pertinence vis à vis d'une requête donnée. Cette classification doit être faite sur la base d'un ensemble de critères documentant le contenu et la qualité des sources et appelés métadonnées.

À partir de la hiérarchie obtenue suite à la classification des sources, nous devons être capable d'extraire les sources susceptibles de répondre au mieux à une question donnée. La méthode d'interrogation des sources doit en outre, prendre en compte la sémantique des requêtes qu'elle

traite pour améliorer les résultats de la recherche.

Organisation du mémoire

Dans le premier chapitre, nous commençons par présenter le formalisme des treillis de Galois ainsi que les principaux travaux sur son application à la recherche d'information. Nous exposerons ensuite quelques propositions concernant l'utilisation des ontologies de domaines dans les systèmes de recherche d'information. Nous évoquons par la suite deux approches incluant des modèles de traitement de requêtes (raffinement, réécriture ...).

Ensuite, nous allons détailler, dans le deuxième chapitre, la méthode que nous avons adoptée pour la classification et la localisation des ressources génomiques pertinentes sur la base d'un ensemble de métadonnées pour des requêtes posées par des utilisateurs. Cette méthode consiste à construire un treillis de Galois classifiant l'ensemble de ces ressources puis à y insérer la requête utilisateur pour extraire par la suite une liste de ressources triées par ordre de pertinence décroissant par rapport à la requête utilisateur.

Dans le dernier chapitre, nous détaillerons l'enrichissement sémantique de requêtes à partir des ontologies de domaine. Nous définissons différents modes d'enrichissement et nous illustrons leurs effets sur le résultat final. Nous finirons en positionnant notre proposition et son apport par rapport aux travaux réalisés dans le même domaine avant de décrire la réalisation dont a fait l'objet de ce stage.

Chapitre 1

Étude bibliographique

Introduction

Nous présentons dans ce chapitre un bref tour d'horizon des travaux auxquels nous avons eu recours pour développer ce travail. Nous introduisons dans un premier temps les notions formelles définissant les treillis de Galois et les algorithmes proposés pour les construire puis nous évoquons les travaux qui ont appliqué les treillis pour la recherche d'information. Nous rappelons ensuite la définition des ontologies de domaine et leur utilisation dans les systèmes de recherche d'information. Nous présentons enfin quelques propositions de traitement (réécriture, raffinement) de requêtes utilisateur en vue d'améliorer leurs exécutions et les résultats retournés.

1.1 Généralités sur les treillis de Galois

Nous ne présentons dans cette section que les notions les plus usuelles relatives aux treillis de Galois nécessaires pour développer les idées introduites dans ce mémoire. Ces notions sont rappelées dans divers travaux étudiant ou utilisant les treillis de Galois [Gue90, GM93, PRSV02]. En revanche, les définitions de base de la théorie mathématique des treillis et les démonstrations qui les accompagnent sont détaillées dans les tout premiers ouvrages qui ont étudié les treillis [Bir67, BM70].

1.1.1 Contexte formel

Un contexte formel est un triplet $K=(S,P,I)$ où S est un ensemble d'objets ou d'individus, P est un ensemble d'attributs ou de propriétés et I est une relation binaire entre S et P vérifiant :

- $I \subseteq S \times P$
- $(s,p) \in I$ avec $s \in S$ et $p \in P$

La relation $(s,p) \in I$ signifie que l'objet s possède l'attribut ou la propriété p.

1.1.2 Connexion de Galois

Soit $\wp(S)$ (respectivement $\wp(P)$) l'ensemble des parties de S (respectivement P). Et considérons les applications suivantes :

$f : \wp(S) \rightarrow \wp(P)$ *définie par* $f(X) = \{p \in P \mid \forall\ s \in X,\ (s,p) \in I\}$

f est l'application qui associe à tout ensemble d'objets de S l'ensemble de leurs attributs communs dans P.

$g : \wp(P) \rightarrow \wp(S)$ *définie par* $g(Y) = \{s \in S \mid \forall\ p \in Y,\ (s,p) \in I\}$

g est l'application qui associe à tout ensemble d'attributs de P l'ensemble des objets de S possédant ces attributs.

Les applications f et g vérifient les propriétés suivantes :
- $\forall$ (S1,S2) $\in \wp$(S), S1 $\subseteq$ S2 $\Rightarrow$ f(S2) $\subseteq$ f(S1)
- $\forall$ (P1,P2) $\in \wp$(P), P1 $\subseteq$ P2 $\Rightarrow$ g(P2) $\subseteq$ g(P1)
- $\forall$ P1 $\in \wp$(P), P1 $\subseteq$ f(g(P1)) et $\forall$ S1 $\in \wp$(S), S1 $\subseteq$ g(f(S1)).

Par définition *(f,g)* forme une connexion de Galois entre *($\wp$(S),$\subseteq$)* et *($\wp$(P),$\subseteq$)*

1.1.3 Fermeture

Une fermeture sur un ensemble ordonné (E,$\leq$) est une application, r : E $\rightarrow$ E, qui, pour tout x, $y \in E$, vérifie les propriétés suivantes :
- $x \leq r(x)$ (r est extensive)
- si $x \leq y$ alors $r(x) \leq r(y)$ (r est monotone croissante)
- $r(x) = r(r(x))$ (r est idempotente)

Un élément x de E est fermé pour r si et seulement si x=$r(x)$.
Les applications $h = f \circ g$ et $h' = g \circ f$ sont respectivement des fermetures sur *($\wp$(S),$\subseteq$)* et *($\wp$(P),$\subseteq$)*. Et les fermés de h (respectivement h') sont les éléments $A \in \wp(S)$ (respectivement $B \in \wp(P)$) tels que $h(A)=A$ (respectivement $h'(B)=B$).

1.1.4 Treillis de Galois

Treillis

Un treillis (E,$\leq$) est un ensemble ordonné tel que chaque couple d'éléments (x,y) possède une borne *sup (supremum)* noté $x \vee y$ et une borne *inf (infimum)* noté $x \wedge y$.

Soient L_S et L_P les ensembles des fermés respectifs pour h et h'. (L_S,$\subseteq$) est le treillis des fermés pour h et(L_P,$\subseteq$) est le treillis des fermés pour h' et (L_S,$\subseteq$) et (L_P,$\subseteq$) sont deux treillis isomorphes.

Concept formel

Pour tout A $\subseteq$ S on définit A' = f(A) = { p $\in$ P / $\forall$ s $\in$ A, (s,p) $\in$ I } et pour tout B $\subseteq$ P on définit B' = g(B) = { s $\in$ S/ $\forall$ p $\in$ B , (s,p) $\in$ I }.
Un concept formel ayant pour contexte formel (S,P,I) est défini comme étant un couple C=(A,B) avec A $\subseteq$ S, B $\subseteq$ P, A'=B et B'=A. Les ensembles A et B sont appelés respectivement *extension* et *intension* du concept formel C.

Treillis de Galois

Considérons L_S et L_P définis précédemment et soit $L = L_S \times L_P$. On définit l'opérateur de subsomption de concept formel, qu'on notera $\preceq$, par :

$$C1=(A1,B1) \preceq C2=(A2,B2) \Leftrightarrow A1 \subseteq A2 \ (\Leftrightarrow B2 \subseteq B1 \ )$$

On dit que *C1 est subsumé par C2* ou encore *C1 est plus spécifique que C2* ou *C2 est plus général que C1*.

(L,$\preceq$) est le produit des deux treillis isomorphes *(L_S,$\subseteq$)* et *(L_P,$\subseteq$)* appelés respectivement *treillis des extensions* et *treillis des intensions*. *(L,$\preceq$)* est par définition *le treillis de Galois* associé à la relation binaire I sur $S \times P$. L'ensemble de tous les concepts formels du contexte *K = (S,P,I)*

muni de l'ordre partiel $\preceq$ est un treillis complet appelé *treillis de concepts de K* ou *treillis de Galois*.

Algorithmes de construction des treillis de Galois

La construction du treillis de Galois d'une relation binaire donnée peut être décomposée en trois parties [GM93] :

1. l'énumération des rectangles maximaux (les fermés),
2. la recherche de la relation d'ordre partiel entre ces rectangles
3. et la représentation du treillis (construction du diagramme de HASSE correspondant au treillis appelé aussi graphe de couverture).

Plusieurs algorithmes ont été proposés pour la construction de hiérarchies de classes à partir de la spécification de leurs propriétés [Gue90]. On ne retiendra de ces algorithmes que ceux qui permettent une mise à jour incrémentale tout en conservant la structure du treillis ainsi que le diagramme de HASSE correspondant. Parmi les algorithmes proposés, seuls deux permettent l'ajout d'un nouvel élément à un treillis déjà construit tout en conservant sa structure [GMM95a, CR93] et un troisième plus récent permet la fusion de deux treillis en un seul [VM01].

1.2 Treillis de Galois et Recherche d'Information

Plusieurs méthodes de classification conceptuelle basées sur les treillis de Galois ont été utilisées dans diverses applications [GMM95b] notamment pour la recherche documentaire où chaque concept du treillis généré à partir d'une relation d'indexation correspond à un ensemble de documents décrits par les termes d'index communs. Dans la perspective de la recherche booléenne, chaque concept peut être vu comme une requête formée de la conjonction des termes d'index du concept (les éléments de son intension). Ainsi, en se basant sur la représentation de la collection de documents dans un treillis de Galois, la recherche documentaire bénéficie de la combinaison de deux modes d'interaction dans un même espace de recherche. En effet, le graphe (treillis de Galois) représente une relation de généralisation/spécialisation entre les requêtes pouvant être satisfaites par les documents de la collection. La recherche est effectuée par une combinaison libre de :

1. la spécification directe de termes d'index, résultant en un saut dans le concept le plus général incorporant les termes spécifiés et les termes du concept de départ,
2. et la navigation libre en suivant les arcs du graphe du treillis.

Le parcours d'un arc correspond à un élargissement (généralisation) ou un raffinement (spécialisation) minimal par rapport à la requête correspondant au concept courant.

L'approche CLR (Concept Lattices-Based Ranking) [CR00] est l'une des propositions les plus intéressantes utilisant les treillis de Galois pour la recherche documentaire. Elle répond plus précisément aux problèmes de vocabulaire (polysémie, synonymie) créés lors de l'utilisation dans les requêtes de termes qui ne correspondent pas forcément aux termes utilisés par les auteurs pour décrire leurs documents. Elle consiste à exploiter les relations entre les documents de la collection dans le contexte existant, lors du choix de ceux qui seront retournés en réponse à la requête et ce à travers la construction d'une structure de cluster à partir de l'ensemble de documents. Une telle construction s'appuie sur la reconnaissance et l'ordonnancement d'un ensemble de relations d'inclusion entre les termes décrivant les documents. L'approche CLR adopte ainsi une nouvelle stratégie de classification consistant à partir d'une représentation groupée de toute la

collection pour diriger une transformation entre la représentation d'une requête et celle de chaque document.

Les principales motivations qui ont amené à cette proposition sont les suivantes :

1. La transformation classique requête-document (méthode vectorielle) peut être vue comme un ensemble d'opérations pour transformer les termes du vecteur représentant la requête en termes du vecteur représentant le document. Cette méthode ne peut pas classer les documents dont les vecteurs ne partagent aucun terme avec celui de la requête.
2. La transformation requête-document doit être conduite par une structure conceptuelle interne de la Base de Données exploitée (la collection de documents).

L'idée inspirée de ces deux problèmes est d'extraire, de chaque document, un ensemble de concepts qui peuvent être des requêtes satisfaites par le document et de définir pour chaque concept un opérateur de voisinage permettant de transiter aux concepts voisins et par suite d'une requête à l'autre. De cette façon, le problème de comptage de la séquence d'opérations de la transformation requête-document est ramené à une recherche en largeur dans l'espace des requêtes admissibles (l'ensemble des concepts extraits). On considère à l'état initial le concept correspondant à la requête et aux états suivants les concepts donnés par l'opérateur voisinage. L'état final est atteint lorsqu'un concept représentant le document est atteint. En parallèle avec cette recherche une attribution de scores aux documents est effectuée. Le score d'un document est donné par le plus court chemin entre l'état initial et l'état final. La classification finale est obtenue en organisant les documents dans l'ordre croissant de leurs scores.

Pour mettre en œuvre cette idée et effectuer la représentation conceptuelle d'une collection de documents qui répond à ces besoins, le treillis de Galois associé à la relation binaire termes×documents semble être un bon candidat. En effet :

- Un concept est une interprétation naturelle du point de vue caractérisation de l'ensemble de requêtes pouvant être satisfait par la collection.
- Une requête utilisateur peut logiquement être insérée dans le treillis.
- L'ordre des concepts montre comment passer d'une requête à l'autre.
- L'ensemble de concepts dans le treillis est suffisamment grand pour assurer à une représentation riche de rester gérable.

En plus de ces éléments, la relation de subsomption entre les concepts du treillis détermine la relation de voisinage et la fermeture transitive identifie une séquence minimale de raffinement ou d'élargissement permettant de dériver un concept à partir d'un autre.

L'implémentation de cette approche consiste à :

- construire le treillis de Galois correspondant à la collection de documents en utilisant le système GALOIS [CR93] pour la construction incrémentale des treillis de Galois,
- insérer la requête dans ce treillis (faisable grâce à la construction incrémentale du treillis)
- et effectuer une recherche en largeur sans compter les nœuds déjà visités. La distance à (ou encore la position de) chaque nœud est comptée lors de sa visite.

1.3 Ontologies de domaines et Recherche d'Information

1.3.1 Définition

Une ontologie est "*une spécification explicite et formelle d'une conceptualisation faisant l'objet d'un consensus* [Gru93]". Elle réunit à la fois des éléments, concepts ou mots, et des règles permettant de manipuler ces éléments ou d'effectuer un certain nombre d'inférences [BLHL01].

La manipulation des concepts ou éléments de l'ontologie est guidée par l'ensemble des caractéristiques (propriétés) qui leur sont attachées et l'ensemble des relations qui définissent la structuration de l'ontologie. Celui-ci contient essentiellement la relation de subsomption "*is-a*" définissant le lien de généralisation entre concepts et "*choisie comme relation de structuration de l'arborescence ontologique*" [CBT03] mais aussi d'autres relations permettant d'unir les concepts pour la construction d'une représentation conceptuelle plus complexe.

1.3.2 Construction et utilisation des ontologies

Suite à l'importante évolution de la recherche autour du le Web sémantique, les systèmes de recherche d'informations ont de plus en plus recours à l'utilisation d'ontologies de domaines pour orienter les choix des sources d'informations à retourner en réponse aux questions (requêtes) des utilisateurs. En l'absence d'une méthode standard pour la recherche d'informations guidée par les ontologies de domaines, plusieurs approches ont été proposées. La mise en place de l'ontologie relative au domaine d'application du système conçu diffère d'une approche à l'autre. En effet plusieurs approches reposent sur une ou plusieurs ontologies construites au préalable alors que d'autres essaient de construire leurs propres ontologies au fur et à mesure que le système exploite de nouvelles informations.

La première solution est adoptée lors de l'évolution d'un système de recherche d'information existant en un système qui tient compte de la sémantique des informations traitées [SQ02]. Cette transformation nécessite d'abord la création de l'ontologie de domaine appropriée puis l'enrichissement du texte par des informations sémantiques pour arriver enfin à la construction du moteur de recherche sémantique. La création de l'ontologie consiste à définir le domaine de connaissance puis le language sémantique capable de représenter l'ontologie et compatible avec le web et enfin les objet structuraux et sémantiques inférés respectivement de la structure et du contenu des documents.

L'utilisation d'ontologies déjà mises en place est adoptée lors de la personnalisation d'un système de recherche d'information en se basant sur les profils utilisateurs et l'historique de leur navigation [CG04].

La construction d'ontologies au fur et à mesure de la réponse aux requêtes est la solution adoptée lors de la création d'un nouveau système de recherche d'information guidée par une ontologie [SQ03]. Elle consiste à effectuer une analyse syntaxique et des transformations XML sur du texte en langage naturel pour obtenir un ensemble de fichiers XML. Une analyse sémantique de ces fichiers permet par la suite d'extraire un ensemble d'expressions logiques du premier ordre. L'ontologie est enfin construite après le raffinement manuel des entités de base extraites de ces expressions logiques. Une fois l'ontologie construite, on procède à l'extraction des instances de ses classes pour les associer avec les documents. Ceci permet d'avoir l'ensemble de documents classés selon l'ontologie.

La combinaison des deux modes de construction d'ontologies à utiliser par les systèmes de recherche d'information est aussi adoptée par plusieurs propositions [MS01]. Elle consiste à importer et réutiliser les ontologies existantes auxquelles seront ajoutées de nouvelles ontologies construites à partir du domaine d'application. Une nouvelle ontologie est construite à partir d'une hiérarchie, établie entre les concepts extraits d'un ensemble d'unités lexicales, et d'un ensemble de règles d'association générées par un algorithme de découverte de propriétés entre classes (concepts). Il est possible d'effectuer un raffinement de l'ontologie construite en utilisant des données de l'application telles que l'historique et les profils utilisateurs.

1.4 Réécriture de requêtes

La réécriture ou l'affinement de requêtes est souvent rencontrée dans les systèmes de médiation qui cherchent à intégrer des sources d'informations préexistantes et généralement réparties (distribuées) et hétérogènes. Ces systèmes fournissent une interface de requête uniforme pour plusieurs sources hétérogènes et autonomes dans le but de permettre le traitement de requêtes complexes. A cet effet, ces systèmes doivent permettre :

- d'identifier les sources pertinentes pour la requête,
- de décomposer la requête complexe en requêtes élémentaires adressables à ces sources,
- d'exécuter ces requêtes,
- et de regrouper (combiner) les résultats retournés par les différentes sources pour former une réponse globale à la requête de l'utilisateur.

Nous présentons brièvement dans ce qui suit deux travaux parmi ceux les plus récents qui présentent sur des systèmes de médiation incluant des modules d'affinement ou de réécriture de requêtes.

Réécriture de requêtes en utilisant des vues

La réécriture de requêtes est étudiée dans le cadre des bases de données relationnelles. La problématique est, étant donné un ensemble de vues $\{v_i\}$ définies sur les relations d'une base de donnée et une requête Q, d'évaluer la possibilité de répondre à Q en utilisant les v_i, de déterminer l'ensemble maximal de tuples dans la réponse à Q qui peuvent être obtenus à partir des v_i et de déterminer le plan d'exécution le moins coûteux pour répondre à la requête [Hal01]. L'enjeu n'est donc pas uniquement de détecter si une vue v_i est utilisable pour répondre à une requête mais aussi de pouvoir décider selon le coût (en temps d'exécution) lors du choix de l'utilisation des vues disponibles.

Cette problématique est rencontrée dans deux classes d'applications. La première est l'optimisation de requêtes et la conception de bases de données où il s'agit de produire plan d'exécution de requêtes qui implique des vues. La deuxième est l'intégration de données où l'on procède à des transformations des requêtes formulées en terme de schéma médiateur en requêtes formulées en terme de sources de données. Dans ce dernier cas, vu que les sources ne couvrent pas toujours tout le domaine, il est possible d'accepter une sous-requête au lieu d'une requête équivalente et ceci permet d'avoir un sous-ensemble du résultat au lieu d'un résultat vide.

Plus concrètement, la réécriture d'une requête Q en utilisant un ensemble de vues $\{v_i\}$ est une requête Q_E qui se réfère uniquement aux v_i. La réécriture est dite équivalente lorsque la requête Q_2 est équivalente à Q_E. En revanche, s'il n'est pas possible d'avoir une réécriture équivalente, une réécriture contenue maximale de Q_E est acceptée.

Informellement une vue est utilisable pour la réécriture d'une requête si les ensembles de relations qu'elle mentionne se chevauchent et si la vue inclut un ensemble d'attributs sélectionnés par la requête. De ce fait si une requête applique des prédicats sur les attributs qu'elle partage avec la vue, celle-ci doit appliquer des prédicats équivalents ou moins stricts sur les mêmes attributs pour qu'elle fasse partie d'une réécriture équivalente. Autrement (si elle applique des prédicats plus stricts) elle peut faire partie d'une réécriture continue dans la réécriture équivalente.

Le projet PICSEL

PICSEL [BFG+02] est un environnement déclaratif de construction de médiateur qui aborde la problématique d'intégration des sources d'informations préexistantes (distantes et hétérogènes). Il intègre un module d'affinement de requêtes qui constitue une brique d'un module

de dialogue coopératif entre le médiateur et ses utilisateurs. Le médiateur, étant dépourvu de données, ne peut pas évaluer directement les requêtes qui lui sont posées. Il procède donc à leur réécriture en termes de vues pour le calcul des plans de requêtes à exécuter afin d'obtenir l'ensemble des réponses à une requête globale. L'affinement de requêtes est déclenché lorsque l'ensemble des plans de requêtes calculé par l'algorithme de réécriture de requêtes en terme de vues est vide. Pour la représentation et la modélisation des connaissance dans PICSEL, un formalisme (CARIN-ALN) a été défini pour être adopté par la suite comme support de description de l'ontologie composée de l'ensemble des prédicats modélisant le domaine d'application du système (langage de vues) et comme langage de requêtes.

Calcul des plans de requêtes dans PICSEL

Le plan de requête qui résulte de sa réécriture en terme de vues n'est autre qu'une requête conjonctive sur les atomes-vues ayant une expansion satisfiable et implique le corps de la requête initiale. Les étapes du calcul du plan de requête sont les suivants

1. La verification de la satisfiabilité de la requête après sa normalisation. Si la requête est insatisfiable alors la procédure de raffinement est déclenchée.
2. Le dépliement de la requête.
3. La réécriture atome par atome de chaque requête conjonctive du déplié.

Affinement de requêtes dans PICSEL

Il y a deux causes de déclenchement de l'affinement de requêtes. Lorsqu'une requête est insatisfiable et cause donc la violation de contraintes d'intégrité du domaine et lorsqu'une requête est non réécrivable à partir des sources disponibles et sort donc des compétences des sources du système. Pour chacun des cas il existe un module qui traite le problème. Il s'agit dans le premier cas de détecter d'abord le conflit et l'atome du corps de la requête qui est à l'origine de l'insatisfiabilité et remplacer celui-ci par sa généralisation pour éviter les contradictions (l'inférence de $\perp$) puis d'effectuer la réparation de la requête qui consiste à considérer les concepts les plus spécifiques de la hiérarchie qui permettent d'avoir une requête satisfiable sans trop s'éloigner de la requête initiale (de l'utilisateur). Dans le cas où la requête est non réécrivable, le système propose à l'utilisateur une requête qui peut être réécrite dans le vocabulaire des sources disponibles et sémantiquement plus proche de la requête initiale. Un troisième type de raffinement qui n'est dû à aucune anomalie dans la requête de l'utilisateur existe aussi. Il s'agit du cas où la requête, bien formulée, ramène trop de réponses. Dans un tel cas le raffinement se ramène à une spécialisation de la requête tout en tenant compte des préférences de l'utilisateur pour ne lui retourner qu'un sous ensemble du résultat.

Conclusion

Dans ce chapitre nous avons présenté les travaux essentiels sur l'application des treillis de Galois à la recherche d'informations, sur l'utilisation des ontologies de domaines pour la recherche d'information et sur la réécriture de requêtes dans les systèmes de médiation. Dans les chapitres suivants nous présenterons une méthode combinant ces techniques, destinée à la classification et la recherche des ressources génomiques susceptibles de répondre à une question donnée.

Chapitre 2

Utilisation des treillis de Galois pour la classification et la recherche des sources biologiques

Introduction

La bioinformatique est un domaine en plein essor dont l'évolution récente se manifeste entre autres par le grand nombre de sources de données (banques de données) disponibles sur le web. Mais, en l'absence d'un standard de description et d'un schéma unique à ces sources, la localisation d'une source susceptible de contenir des informations précises demandées par un utilisateur, notamment un biologiste, n'est pas toujours une tâche évidente, d'où la sous-exploitation de ces données. De ce fait, établir une classification des sources reflétant leur contenu et facilitant leur localisation s'avère incontournable. Pour établir une telle classification, les treillis de Galois semblent un bon candidat permettant d'organiser les sources en une hiérarchie de classes où chaque classe correspond à un ensemble de sources partageant un ensemble de propriétés communes qui reflètent leur contenu et leur qualité et peuvent servir de clés pour la localisation de ces sources.

Nous commençons ce chapitre par introduire le catalogue des sources à partir duquel nous définissons le contexte formel. Celui-ci servira pour la construction du treillis de Galois que nous présentons par la suite. Ensuite nous détaillerons la méthode de recherche des sources pertinentes pour une requête donnée. Enfin nous effectuons le bilan de cette méthode en énumérant ses avantages et ses limites.

2.1 Catalogue des sources et contexte formel

2.1.1 Catalogue des sources

Depuis quelques années la plupart des banques de données biologiques sont disponibles sur le réseau Internet. Ceci leur a permis d'offrir aux utilisateurs un accès plus facile aux données notamment grâce aux interfaces web permettant à la fois un mode d'interrogation simplifié et une présentation satisfaisante des résultats. Cependant, ces banques de données, étant d'une part mises en place par différents acteurs et d'autre part destinées à divers domaines biologiques, n'ont pas forcément le même schéma ni le même mode (et interface) d'interrogation. Cette hétérogénéité de schémas et de contenus des banques de données biologiques découle de la richesse

et de la prolifération des informations biologiques disponibles sur le web. Toutefois cette même hétérogénéité est à l'origine de la difficulté d'obtenir, pour une requête donnée, des informations complètes répertoriées sur plusieurs banques. Pour remédier à ce problème, un mode commun d'interrogation ou un système de médiation doit être mis en place. Les systèmes SRS[1] et NCBI[2] sont des exemples d'accès unifié à un grand nombre de sources hétérogènes. Ils s'appuient sur l'existence de catalogues répertoriant les sources et facilitant leur accès. Parmi ces catalogues figure le catalogue DBCAT[3] d'INFOBIOGEN [4]. DBCAT est un catalogue exhaustif et détaillé de la majorité des banques de données biologiques. Il contient 509 entrées (au 1^{er} mars 2004). Chaque entrée est structurée en plusieurs champs qui constituent une sorte de profil de la banque de données correspondante. La possibilité d'interrogation et la richesse du contenu font de DBCAT un outil très intéressant. Toutefois DBCAT est marqué par l'absence de maintenance et par suite l'absence de mise à jour des informations sur les banques de données référencées. Il était donc préférable dans le cadre de notre projet de créer notre propre catalogue des sources biologiques dans lequel nous ne gardons des entrées de DBCAT que les plus pertinentes. Celles-ci seront par la suite enrichies avec des métadonnées supplémentaires relatives au contenu et à la qualité des banques de données référencées. La mise en forme des entrées du catalogue reste, dans le cadre de ce travail, une tâche manuelle mais une perspective est de mettre en place une technique de fouille de l'ensemble des sites Web des banques de données pour alimenter le catalogue en fonction des mises à jour.

L'exploitation du catalogue vise à atteindre un double objectif : la classification et la recherche des sources biologiques. En effet l'un des problèmes fréquemment rencontrés par l'utilisateur est de trouver la ou les sources de données pertinentes pour une question posée. Par ailleurs le nombre et l'hétérogénéité des sources rend leur classement très difficile [Gal04], même si certains portails thématiques proposent déjà des classifications selon des points de vue particuliers (exemples portail INFOBIOGEN, BioResearch[5]).

C'est dans ce contexte que nous avons choisi de développer et tester une méthode s'appuyant sur les treillis de Galois permettant à la fois la classification et la recherche des sources de données biologiques ainsi que la prise en compte de la sémantique du domaine.

2.1.2 Contexte formel

On reprendra ici la définition du contexte formel *K=(S,P,I)* introduite dans la section 1.1.1 en considérant que S désigne l'ensemble des sources et P désigne l'ensemble des propriétés. Le contexte formel K sera donc présenté par un tableau *sources* $\times$ *propriétés* où, toute case du tableau contiendra soit 0 soit 1 selon que le couple *(s,p)* de $S \times P$ appartient à I ou non. L'ensemble S contient toutes les sources figurant dans le catalogue et on retiendra dans l'ensemble P les propriétés pouvant servir comme critères utilisables plus tard pour la localisation des sources pertinentes dans les requêtes des utilisateurs.

Exemple

Le tableau 2.1 présente un exemple de contexte formel. Il présente un ensemble de sources biologiques avec les propriétés qu'elles possèdent. Par exemple la deuxième ligne traduit le fait que la source *Swissprot* possède les propriétés *Séquence protéique*, *Tout Organisme* et *Annotation*

[1] http ://www.infobiogen.fr/srs6bin/cgi-bin/wgetz ?-page+LibInfo+-lib+DBCAT+-newId
[2] http ://www.ncbi.nlm.nih.gov/
[3] http ://www.infobiogen.fr/services/dbcat/
[4] Centre de Ressources INFOBIOGEN 2003, http ://www.infobiogen.fr
[5] http ://bioresearch.ac.uk/

Gene Ontology. Ceci exprime que *Swissprot* comporte des séquences protéiques relatives à tous les organismes et que les données qu'elle contient sont annotées selon l'ontologie *Gene Ontology*.

Pour une représentation plus significative, les noms complets des propriétés (non représentés dans le tableau 2.1 faute d'espace) sont donnés par le tableau 2.2.

Sources\Propriétés	SN	SP	TO	An	Ve	Ho	So	AGO
Swissprot	0	1	1	0	0	0	0	1
RefSeq	1	1	1	0	0	0	0	1
TIGR-HGI	1	0	0	0	0	1	0	1
GPCRDB	0	1	1	0	0	0	0	0
HUGE	1	1	0	0	0	1	0	0
ENSEMBL	1	0	0	1	0	0	0	0
Mouse Genome DB	0	1	0	0	0	0	1	1
Vega Genome Browser	0	1	0	0	1	0	0	0

TAB. 2.1 – Exemple de contexte formel (K=(S,P,I)).

Abréviation	Source
S1	Swiss Prot
S2	RefSeq
S3	TIGR-HGI
S4	GPCRDB
S5	HUGE
S6	ENSEMBL
S7	Mouse Genome DB
S8	Vega Genome Browser

Abréviation	Propriété
SN	Séquence Nucléique
SP	Séquence Protéique
TO	Tout Organisme
An	Animaux
Ve	Vertébrés
Ho	Homme
So	Souris
AGO	Annotation Gene Ontology

TAB. 2.2 – Noms complets des sources biologiques et de leurs propriétés.

2.2 Construction incrémentale du treillis de Galois

Plusieurs algorithmes de construction de treillis de Galois ont été proposés parmi lesquels figure "Incremental Structuring of Knowledge Bases" [GMM95a]. Le choix de cet algorithme repose sur le fait qu'il offre la possibilité d'ajout de nouveaux concepts à un treillis déjà construit. En effet, il consiste en une comparaison des attributs du nouveau concept à insérer avec une recherche par spécialisation dans le treillis, en construisant des nouveaux noeuds si nécessaire.

La figure 2.1 représente le treillis de Galois correspondant au contexte formel K donné par la tableau 2.1. On notera $\Theta(K)$ ce treillis.

2.3 Recherche des sources pertinentes

Après avoir construit le treillis de Galois, on peut commencer la recherche des sources pertinentes par rapport aux propriétés souhaitées par l'utilisateur et exprimées dans sa requête. Pour ce faire, il s'agit d'abord de représenter la requête sous la forme d'un concept (A_Q, B_Q) où A_Q est l'ensemble formé d'un seul élément *Query* et B_Q est l'ensemble formé par les propriétés

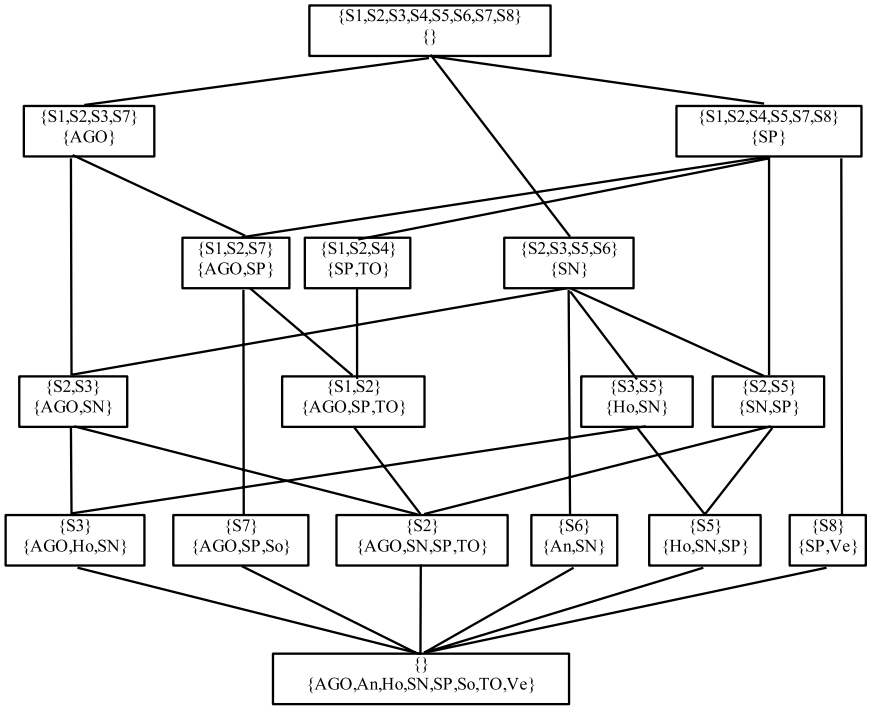

FIG. 2.1 – **Treillis de Galois $\Theta(K)$ correspondant au contexte K**

figurant dans la requête de l'utilisateur. Il est à noter ici que B_Q peut ne pas être un fermé dans S. Ceci est vrai lorsque l'ensemble de propriétés de la requête est possédé par une ou plusieurs sources ce qui se traduit dans le treillis de Galois par l'existence d'un concept tel que l'intension est égale à B_Q. Dans ce cas le couple (A_Q, B_Q) n'est pas un concept Galois.

Il s'agit ensuite d'ajouter ce concept au treillis de Galois déjà construit. Cet ajout est possible et préserve la structure du treillis grâce à l'utilisation de l'algorithme de construction incrémentale de treillis de Galois [GMM95a] précédemment mentionné. L'étape suivante consiste à chercher, dans le treillis résultant, le concept $C_R=(A_R,B_R)$ tel que l'intension B_R soit l'ensemble de cardinalité minimale contenant l'ensemble B_Q (l'ensemble des propriétés exprimées dans la requête).

Proposition 2.1 *Soit $C_Q = (A_Q,B_Q)$ le concept requête et soit $C_R=(A_R,B_R)$ le concept du treillis de Galois vérifiant $|B_R| = min\{|B|, C=(A,B) \in \Theta(K)$ tel que $B_Q \subseteq B\}$ qui traduit que C_R est le concept ayant l'intension de cardinalité minimale contenant les propriétés de la requête, alors les deux ensembles B_R et B_Q sont égaux.*

Preuve 2.1 *Par définition de C_R on a $B_Q \subseteq B_R$ et B_R est l'ensemble de cardinalité minimale contenant B_Q.*

Supposons que $B_R \backslash B_Q \neq \emptyset$. Ceci est équivalent à $|B_R| \geq |B_Q|+1$ (1) et donc $|B_R| > |B_Q|$. $B_R \backslash B_Q \neq \emptyset$ implique que A_Q n'est pas inclus dans A_R car $B_R \backslash B_Q$ n'est pas inclus dans B_Q et par suite les propriétés de cet ensemble ne sont pas dans la requête.

Par définition du treillis, il existe un concept $C_S=(A_S,B_S)$, le supremum *de C_R et C_Q, tel que $A_S = A_R \cup A_Q$ et $B_S=B_R \cap B_Q = B_Q$ (2) (car $B_Q \subseteq B_R$).*

Les relations (1) et (2) donnent $|B_S|<|B_R|$. Ceci traduit l'existence d'un concept ayant une

intension qui contient $|B_Q|$ et de cardinalité inférieure à celle de B_R ce qui contredit la définition de C_R. On déduit ainsi que $B_R \backslash B_Q \neq \emptyset$ et par suite $B_Q = B_R$.

Proposition 2.2 *C_R est l'unique concept du treillis ayant B_Q comme intension.*

Preuve 2.2 *La preuve de cette proposition découle directement de la propriété de fermeture des extensions des concepts du treillis dans S.*

Proposition 2.3 *Les sources pertinentes par rapport à la requête de l'utilisateur sont celles figurant dans l'ensemble construit par l'union de l'extension du concept C_R et des extensions de ses subsumants ayant une intension non vide.*

Preuve 2.3 *Par définition du concept $C_R=(A_R,B_R)$, $B_Q \subseteq B_R$, ce qui traduit le fait que toutes les sources figurant dans A_R possèdent les propriétés de la requête.*

Soit C=(A,B) un subsumant de C_R d'intension non vide.

$C_R \preceq C$ est équivalent à $B \subseteq B_R$ ce qui exprime que toutes les propriétés dans B sont des propriétés de la requête. Ajoutons à ceci la condition $B \neq \emptyset$, nous déduisons que toute source figurant dans C partage au moins une propriété avec la requête, d'où sa pertinence.

Stratégie de recherche dans le treillis

L'extension A_R du concept C_R contient au moins l'élément *Query*. À ce niveau, une première étape consiste à ajouter au résultat à retourner (encore vide) les éléments de $A_R \backslash \{Query\}$. Notons R_0 cet ensemble de réponses initiales de rang 0, égal à celui du concept requête dans le treillis. L'étape suivante consiste à considérer l'ensemble des concepts parents directs de C_R dans le treillis, appelés aussi subsumants les plus spécifiques de C_R. Notons $\mathcal{C}_1$ cet ensemble, pour chaque concept $C_{\mathcal{C}_{1i}} = (A_{\mathcal{C}_{1i}}, B_{\mathcal{C}_{1i}})$ de $\mathcal{C}_1$, on construit l'ensemble des réponses de rang 1 par

$R_1 = \bigcup_{i=1..|\mathcal{C}_1|} \{(A_{\mathcal{C}_{1i}} \backslash R_0)$ pour tout $A_{\mathcal{C}_{1i}}$ intension de concept $C_{\mathcal{C}_{1i}} = (A_{\mathcal{C}_{1i}}, B_{\mathcal{C}_{1i}})$ tel que $B_{\mathcal{C}_{1i}} \neq \emptyset\}$. $|\mathcal{C}_1|$ étant le cardinal de $\mathcal{C}_1$.

En effet, l'extraction des réponses n'a pas de sens à partir du concept parent le plus général $C_G = (A_G, B_G)$ tel que $B_G = \emptyset$, et $A_G = S$, l'ensemble complet des sources, puisque par définition ce concept donnera en réponse la totalité des sources du contexte formel. Pour continuer, il faudra considérer $\mathcal{C}_2$, l'ensemble des concepts parents-directs des concepts de $\mathcal{C}_1$. Les nouveaux résultats ajoutés à cette étape seront extraits de chaque extension $A_{\mathcal{C}_{2i}}$ des concepts $C_{\mathcal{C}_{2i}}$ de $\mathcal{C}_2$, après soustraction de l'ensemble des réponses déjà fournies c'est à dire ici $R_0 \cup R_1$. L'itération récursive de rang n s'écrira donc :

$R_n = \bigcup_{i=1..|\mathcal{C}_n|} \{A_{\mathcal{C}_{ni}} \backslash \bigcup_{j=1..n-1} \{R_j\}$ pour tout $A_{\mathcal{C}_{ni}}$ intension de concept $C_{\mathcal{C}_{ni}} = (A_{\mathcal{C}_{ni}}, B_{\mathcal{C}_{ni}})$ tel que $B_{\mathcal{C}_{ni}} \neq \emptyset\}$.

Du fait de la condition sur les extensions des concepts réponses, l'itération s'arrête forcément lorsque l'ensemble des concepts parents direct de rang n ($\mathcal{C}_n$) est égal au concept subsumant le plus général $C_G = (A_G, B_G)$ tel que $A_G = S$ et $B_G = \emptyset$. Dans ce cas il n'y a pas d'ensemble $A_{\mathcal{C}_{ni}}$ distinct de S et R_n est vide. Une autre condition d'arrêt mettant en jeu le nombre total de sources à retourner dans le résultat final, ou le nombre d'étapes à effectuer peut être facilement ajoutée à cet algorithme. En effet on peut fixer une cardinalité maximale de l'ensemble résultat et à la fin de chaque étape on teste si le nombre de sources qu'il contient a atteint ou dépassé cette valeur auquel cas, on sort de l'algorithme. On notera ici qu'il est toutefois possible de dépasser le seuil fixé pour ne pas introduire des préférences aléatoires sur les sources de même degré de pertinence. Par exemple si le seuil fixé est égal à 10 et si à l'étape n-1 le résultat contient 8 sources et que l'étape n retourne trois nouvelles sources, alors les 11 sources seront retenues et l'algorithme s'arrête à ce niveau.

Exemple

Considérons le treillis représenté dans la figure 2.1. Un exemple de requête utilisateur peut être : *"Homme, Séquence Nucléique, Annotation Gene Ontology"* qui exprime le fait que l'utilisateur cherche une ou plusieurs sources, ordonnées selon le nombre de propriétés qu'elles partagent avec la requête, ayant tout ou une partie des propriétés mentionnées. Le concept construit à partir de cette requête est C_Q=*({Query},{Homme, Séquence Nucléique, Annotation Gene Ontology})*. Celui ci est ensuite inséré dans le treillis. Pour cet exemple, l'insertion de C_Q dans le treillis de la figure 2.1 n'entraîne pas l'ajout de nouveaux concepts. En revanche elle entraîne le changement de certains concepts déjà présents dans le treillis. La figure 2.2 illustre les modifications apportées au treillis et donne une idée sur les étapes d'ajout des sources pertinentes au résultat. Les entiers figurant à côté des concepts modifiés représentent les étapes (itérations) pendant lesquelles les concepts correspondants ont été visités (considérés) lors de la construction du résultat. C'est dans l'ordre donné par ces entiers que les sources figureront dans le résultat final. Ainsi, le résultat retourné en réponse à la requête considérée est constitué des sources des

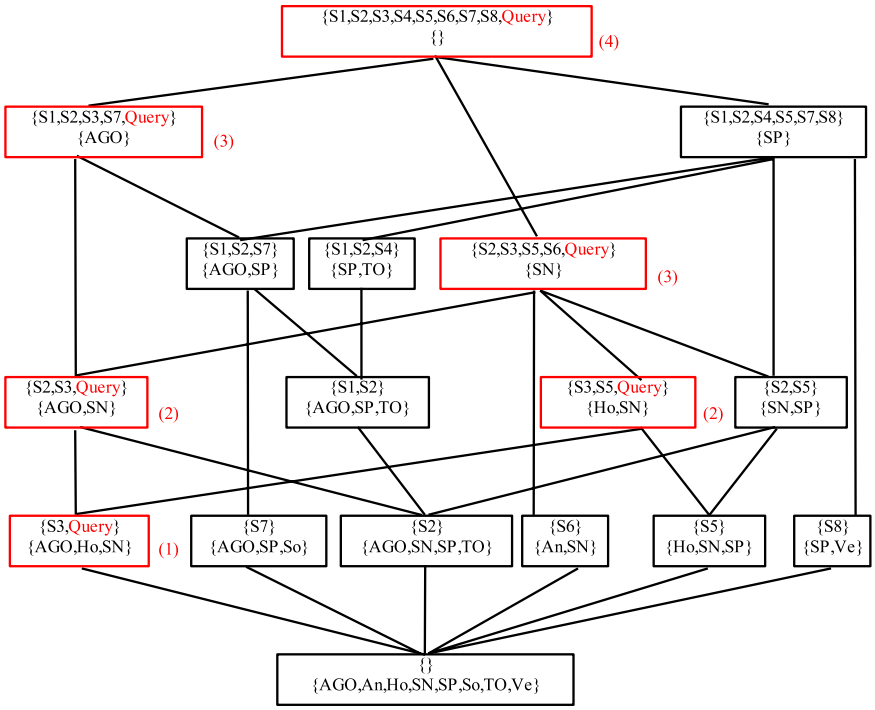

FIG. 2.2 – **Treillis $\Theta(K)$ modifié suite à l'insertion de la requête**

ensembles R_0={S3}, R_1={S2, S5} et R_2={S1, S6, S7}. Il sera donc présenté comme suit :

0. S3 (TIGR-HGI) qui partage avec la requête ses trois propriétés.

1. S2 (RefSeq) qui partage avec la requête les deux propriétés *Séquence Nucléique* et *Annotation Gene Ontology*.

1. S5 (HUGE) qui partage avec la requête les deux propriétés *Séquence Nucléique* et *Homme*.

2. S1 (SwissProt) qui partage avec la requête une seule propriété, *Annotation Gene Ontology*.

2. S6 (ENSEMBL) qui partage avec la requête une seule propriété, *Séquence Nucléique*.

2. S7 (Vega Genome Browser) qui partage avec la requête une seule propriété, *Annotation Gene Ontology*.

Il est à noter ici que l'ordre sur les sources introduites lors d'une même étape (S2 et S5 par exemple) est arbitraire dans cet exemple, mais une heuristique du domaine peut consister à introduire un ordre défini sur la date de la dernière mise à jour des sources.

2.4 Bilan

2.4.1 Avantages de la méthode

L'utilisation du treillis de Galois construit à partir d'un contexte formel sources×propriétés pour la représentation d'une collection de sources biologiques permet d'obtenir une hiérarchie de classes de sources. Cette hiérarchie étant caractérisée par le partage d'un certain nombre de propriétés, permet d'exprimer des relations d'ordre sémantique entre ses classes (nœuds de la hiérarchie). En effet deux types de parcours du treillis sont possibles et traduisent chacun un type de relation entre les nœuds de la hiérarchie. Le parcours par généralisation, qui consiste à passer d'un concept à l'un de ses subsumants, correspond au passage à une classe plus générale où plus de sources possèdent une partie des propriétés de la classe courante. Le parcours par spécialisation, qui consiste à passer d'un concept à l'un de ses subsumés, correspond au passage à une sous classe marquée par une restriction sur l'ensemble des sources due à l'augmentation du nombre de propriétés possédées. En plus de cet aspect sémantique, les classes de la hiérarchie, étant des concepts formels, reposent sur un fondement mathématique permettant d'effectuer non seulement une formalisation simple des résultats offerts par cette hiérarchie mais aussi une preuve de l'exactitude de ces résultats. Les treillis de Galois ont aussi l'avantage de donner une présentation exhaustive de l'ensemble des classes de sources. En effet les concepts représentés dans le treillis correspondent exactement à toutes les classes de sources qu'on peut déduire à partir du contexte formel en question. Cette représentation complète et précise justifie l'exactitude et l'unicité des réponses aux requêtes posées par les utilisateurs.

2.4.2 Limites de la méthode

La hiérarchie de classes de sources donnée par la construction du treillis de Galois exprime des relations sémantiques, dues au partage de propriétés, entre ses classes. Dans la méthode présentée ci-dessus, toutes les propriétés sont considérées comme indépendantes. De ce fait, l'existence possible de relations sémantiques entre les propriétés n'est pas prise en compte. Or des relations sémantiques peuvent exister entre les propriétés d'un sous-ensemble de P du contexte formel $K=(S, P, I)$, notamment lorsque ces propriétés appartiennent à une même ontologie. L'absence de prise en compte des relations sémantiques entre les propriétés constitue une limite de la méthode présentée jusqu'à présent, ainsi que l'illustre l'exemple suivant.

Exemple

Reprenons le treillis de la figure 2.1 et considérons la requête cherchant les sources qui ont les propriétés *Poulet* et *Annotation Gene Ontology*. En suivant la démarche détaillée plus haut, on obtient comme réponses de rang 1 R_1={*S1,S2,S3,S7*}. Ces sources sont trouvées grâce au partage de la propriété *Annotation Gene Ontology*. La propriété *Poulet* qui exprime que l'utilisateur cherche une ou des sources relatives à cet organisme ne permet évidemment pas de trouver de source réponse puisqu'elle n'apparaît pas dans le contexte formel de ce treillis de Galois. Or un examen rapide de ce contexte conduit à penser que d'autres sources non contenues dans *R1*

pourraient intéresser l'utilisateur (par exemple *S6 : ENSEMBL* ou *S8 : Vega Genome Browser*, annotées respectivement par *Animaux* et *Vertébrés* en ce qui concerne les organismes couverts pas ces sources).

Conclusion

Dans ce chapitre nous avons présenté une méthode de classification et de recherche des sources biologiques s'appuyant sur les treillis de Galois. Cette méthode consiste à créer, à partir d'un catalogue des sources, un contexte formel qui servira à la construction du treillis de Galois représentant la hiérarchie des sources. Une fois le treillis construit, l'étape de recherche de sources pertinentes peut commencer. Grâce au mode de classification offert par les treillis de Galois, la méthode permet de n'avoir que des sources pertinentes (ayant au moins l'une des propriétés demandées par l'utilisateur). Toutefois, faute d'informations explicites sur la sémantique des propriétés, la méthode peut passer sous silence des sources parfois plus pertinentes que celles retournées. Ainsi il est indispensable de trouver un moyen d'exprimer les relations sémantiques entre les propriétés puis de les considérer lors de la recherche des sources pertinentes par rapport à une requête.

La représentation et la prise en compte des relations sémantiques susceptibles d'exister entre les propriétés seront détaillées dans le chapitre suivant.

Chapitre 3

Enrichissement sémantique de requêtes

Introduction

L'utilisation des treillis de Galois pour la classification et la recherche des sources pertinentes pour une requête donnée est une méthode de recherche syntaxique qui n'exploite pas les relations sémantiques susceptibles d'exister entre différentes propriétés. De fait dans la plupart des cas ces propriétés sont organisées en arborescence (hiérarchie ou taxonomie) où chaque arête entre deux nœuds de l'arborescence traduit une relation sémantique entre les propriétés représentées par ces deux nœuds.

Dans ce chapitre, nous allons tenir compte des relations sémantiques possibles entres les propriétés pour effectuer l'enrichissement sémantique de la requête utilisateur dans le but d'améliorer le résultat de la recherche. Nous présenterons d'abord un exemple de hiérarchie de propriétés avec une proposition de formalisation puis nous détaillerons la phase d'enrichissement sémantique de la requête. Nous présenterons ensuite les apports de cet enrichissement par rapport aux résultats du chapitre précédent avant de conclure par un positionnement de notre proposition par rapport aux travaux voisins et une brève description de la réalisation.

3.1 Hiérarchie des propriétés

3.1.1 Exemple d'ontologie de propriétés

Parmi les propriétés des sources biologiques, plusieurs constituent des informations complémentaires qui peuvent être regroupées pour former un ensemble complet décrivant une propriété plus générale des sources. Considérons par exemple les trois propriétés *Homme*, *Souris* et *Vertébrés* figurant dans les tableaux 2.2 et 2.1. Ces trois propriétés sont relatives à des organismes étudiés par les sources. Il est clair que les deux premières sont relatives à deux types disjoints d'organismes. En revanche elles sont toutes les deux groupées dans la troisième qui décrit un ensemble d'organismes de façon plus générale. Ainsi il est possible de construire, à partir de chaque ensemble de propriétés liées sémantiquement, une hiérarchie qui permet d'exploiter les différentes relations entre elles.

Les hiérarchies correspondant aux groupes de propriétés peuvent être représentées par des ontologies de domaine qui permettent, en plus de la représentation de la hiérarchie, d'expliciter les relations existant entre les propriétés figurant dans cette hiérarchie. En effet une ontologie correspond à un vocabulaire contrôlé et organisé et à la formalisation explicite des relations créées entre les différents termes du vocabulaire. Les hiérarchies représentées ne contiennent pas

uniquement les propriétés figurant dans le contexte mais aussi des propriétés supplémentaires dans la perspective d'avoir un système plus riche en terme du nombre de sources étudiées. Il s'agit donc d'ontologies plus complètes du domaine qui peuvent servir comme des références solides pour l'enrichissement sémantique qui sera développé plus loin dans ce chapitre.

La figure 3.1 représente une ontologie de domaine extraite de *Tree of life* [6] et *Taxo* [7] en ne retenant que les organismes modèles les plus étudiés en génomique (les feuilles de l'arbre) et les nœuds structurant (le reste des nœuds). Cette ontologie relie les propriétés relatives au type d'organisme couvert par les sources.

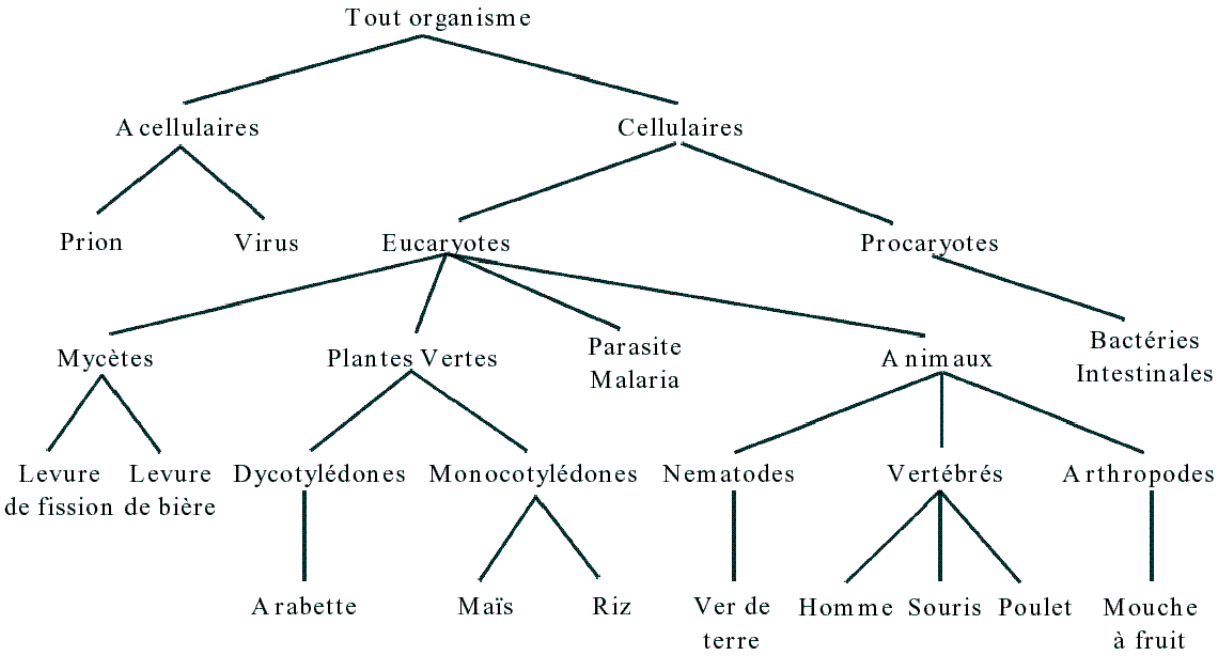

FIG. 3.1 – **Ontologie des organismes modèles**

Remarque 3.1 *Tout ensemble de propriétés représenté par une ontologie est issue d'un ou plusieurs champs de description de sources initialement non booléen. Et comme la construction d'un treillis de Galois exige que les attributs présents dans le contexte formel soient binaires, nous avons procédé à la decomposition de ces champs en propriétés booléennes disjointes. Cette décomposition entraîne la perte des informations indiquant que les nouvelles propriétés sont des valeurs possibles d'un même champ (une même propriété initiale). Pour remédier à cette perte d'informations, des ontologies de domaines rétablissant les relations entre les propriétés séparées sont utilisables.*

3.1.2 Représentation formelle d'une ontologie

Formellement, chaque ontologie qui fait référence à une ou plusieurs propriétés sur les sources est représentée par un arbre qu'on notera $T = (V,E)$ où V est l'ensemble des sommets de T qui représentent chacun une propriété appartenant ou non à l'ensemble P de propriétés définies dans

[6] http ://tolweb.org/tree/phylogeny.html
[7] http ://www.ncbi.nlm.nih.gov/entrez/query.fcgi ?db=Taxonomy

le contexte formel *K=(S,P,I)* et E est l'ensemble des arêtes entre les sommets de V. T est un arbre *enraciné* et sa racine, notée r, est le sommet représentant la propriété multivaluée initiale.

Pour développer l'idée de l'enrichissement sémantique de requêtes, il est nécessaire de rappeler, pour l'arbre T = *(V,E)* de racine r défini précédemment, les notions mathématiques suivantes [CLR92] :

Soient *a,b* $\in$ V, deux sommets de T. On appelle *chemin* entre a et b une suite finie de n sommets (v_i) tels que $a = v_1$, $b = v_n$ et pour tout i dans [1, n-1] il existe une arête dans E entre v_i et v_{i+1}. On notera par *p=*(v_1,v_2, ... ,v_n) un tel chemin et par l = *n-1* sa *longueur*.

Un chemin est dit *élémentaire* si un sommet y est présent au plus une fois (les chemins de T sont tous élémentaires).

On dit que les deux sommets a et b sont *voisins* s'il existe une arête dans E entre a et b.

Le *père* d'un sommet a est l'unique voisin de a sur le chemin de la racine à a. La racine r est le seul sommet sans père.

On appelle *ancêtres* de a tout sommet x qui apparaît sur le chemin de a à la racine r. On dit aussi que a est *descendant* de x.

Les *fils* d'un sommet a sont les voisins de a autres que son père. Une *feuille* est un sommet sans fils.

La hauteur *h(T)* de l'arbre T est la longueur du plus long chemin de la racine à une feuille.

Le sous-arbre de racine a est l'arbre composé des descendants de a enraciné en a.

On appelle *profondeur* d'un sommet a la longueur du chemin de la racine r à a.

Dans un arbre, l'orientation est implicite et se fait par rapport à la racine.

3.2 Enrichissement de la requête

L'enrichissement de la requête de l'utilisateur est l'idée-clé de notre proposition. Il consiste à ajouter à la requête utilisateur de nouvelles propriétés à partir des ontologies de domaine disponibles pour avoir un résultat plus riche. Il s'agit donc, pour une requête donnée, de considérer les propriétés figurant dans l'une des ontologies de domaine. Pour chacune de ces propriétés, on effectue un parcours de l'arbre T correspondant pour en extraire des sommets qui seront ajoutés en tant que propriétés à la requête initiale. On distingue deux types de parcours de T, un parcours par généralisation et un parcours par spécialisation. Ces deux modes reflètent respectivement l'enrichissement par généralisation et l'enrichissement par spécialisation que nous allons présenter dans ce qui suit.

3.2.1 Enrichissement par généralisation

L'enrichissement par généralisation consiste d'abord à localiser le sommet a de T correspondant à l'une des propriétés figurant dans la requête. Puis à parcourir le chemin de a jusqu'à la racine et ajouter à la requête les sommets rencontrés qui appartiennent à l'ensemble P des propriétés dans le contexte formel K. La proposition suivante justifie ce choix.

Proposition 3.1 *Soient* K = (S,P,I) *un contexte formel,* T = (V,E) *l'arbre représentant l'ontologie de domaine considérée. Soit* a *le nœud de* T *correspondant à l'une des propriétés de la requête et soit p=*(v_1,v_2, ... ,v_n) *le chemin de* a=v_1 *à* r *racine de* T. *Parmi les sommets du chemin p, seuls ceux appartenant à* P *sont susceptibles d'augmenter le nombre de sources retournées en réponse à la requête.*

Preuve 3.1 *La preuve de cette proposition est directe à partir du moment ou on sait qu'une propriété n'appartenant pas à* P *ne sera possédée par aucune source de* S. *En plus la considération d'une telle propriété dans l'intension du concept* C_Q=(A_Q,B_Q *à insérer dans le treillis peut entraîner une iteration supplémentaire lors du parcours du treillis pour la comptabilisation du résultat. En effet dans le cas ou* C_Q *n'est pas un concept Galois (l'extension n'est pas un fermé de S), l'ajout d'une propriété ne figurant pas dans* P *à son intension va en faire un concept Galois et du coup la première étape de recherche de sources dans le treillis ne donnera aucune source.*

Ainsi, étant donné une requête ayant initialement l'ensemble de propriétés B_Q et soit $a \in B_Q$ une propriété (nœud) figurant dans une ontologie T. Si on ne fixe pas de contraintes sur le nombre de propriétés à ajouter à une requête, l'ensemble de propriétés de la requête enrichie sera

$$(B_Q \cup \{v_i,\ i=2..n\ /\ p=(v_1,v_2,\ \ldots\ ,v_n)\ \textit{est le chemin de}\ a=v_1\ \textit{à r, racine de T}\}) \cap P.$$

Cet enrichissement est effectué en particulier lorsque le sommet a est une feuille de T et que cette propriété n'apporte pas de sources au résultat (absente de P).

Interprétation

L'enrichissement par généralisation permet d'obtenir une réponse enrichie par des sources plus générales que celles demandées par l'utilisateur vis-à-vis de la propriété appartenant à l'ontologie et figurant dans la requête initiale. Cet enrichissement se base sur le fait qu'une source qui étudie un ensemble d'organismes donnée peut contenir des informations sur l'ensemble plus spécialisé qui interesse l'utilisateur. Par exemple lorsqu'un utilisateur demande une source relative à l'organisme poulet, une source relative aux organismes vertébrés (plus général dans l'ontologie) peut l'intéresser puisqu'elle est susceptible de contenir des informations sur les poulets.

Exemple

Considérons l'exemple introduit à la section 3.2.1, la requête ayant l'ensemble de propriétés B_Q={*Poulet, Annotation Gene Ontology*}. La propriété *Poulet*, étant absente de l'ensemble des propriétés du contexte, ne ramène aucune source. Nous procédons donc à l'enrichissement par généralisation de la requête. Cet enrichissement donne une nouvelle requête ayant en plus de B_Q les propriétés qui figurent dans le chemin de *Poulet* à la racine de l'arbre et qui sont aussi dans P à savoir {*Vertébrés, Animaux, Tout Organisme*}. L'intension de la requête sera donc {*Vertébrés, Animaux, Tout Organisme, Annotation Gene Ontology*}.

3.2.2 Enrichissement par spécialisation

L'enrichissement par spécialisation diffère de celui par généralisation par les sommets de T à ajouter à la requête initiale. En effet, au lieu d'ajouter les *ancêtres* du sommet a dans T qui figurent dans l'ensemble P des propriétés du contexte formel $K=(S,P,I)$ (a est le sommet qui correspond à l'une des propriétés de la requête), nous ajoutons les sommets du sous arbre $T'=(V',E')$ de T enraciné en a qui appartiennent à l'ensemble P. L'ensemble de propriétés de la requête enrichie sera donc

$$(B_Q \cup V') \cap P.$$

Interprétation

Du point de vue biologique, les descendants de a dans T représentent des spécialisations de la propriété représentée par le sommet a. L'ajout de ces propriétés à la requête est dans le but d'enrichir le résultat par des sources répondant à une partie de la requête de l'utilisateur. Par exemple lorsqu'un utilisateur demande une source relative aux eucaryotes, une source relative aux animaux (un sous ensemble des eucaryotes) peut l'intéresser dans le sens ou elle contient des informations sur une partie du groupe d'organismes qu'il demande.

Une contrainte doit être apportée ici au contexte formel à savoir que chaque source ne doit être associée qu'à une et une seule propriété de l'ontologie $T=(E,V)$. Ainsi la réunion des propriétés $(B_Q \cup V') \cap P$ qui est traitée dans le treillis comme conjonction de propriétés se comportera de fait comme si les propriétés appartenant à l'ontologie étaient traitées indépendamment les unes des autres.

Exemple

Considérons la requête où B_Q={*Eucaryotes, Annotation Gene Ontology*}. La propriété *Eucaryotes* figure dans l'ontologie représentée par la figure 3.1. L'enrichissement par spécialisation de la requête donne une nouvelle requête ayant les propriétés données par la formule $(B_Q \cup V') \cap P$ = {*Annotation Gene Ontology, Animaux, Vertébrés, Homme, Souris*}

Les requêtes posées par les biologistes concernent souvent des nœuds feuilles de l'arbre T. Dans de tels cas il n'est pas possible d'effectuer l'enrichissement par spécialisation ce qui justifie un recours plus fréquent à l'enrichissement par généralisation qu'à celui par spécialisation.

Il est à noter qu'un enrichissement mixte est tout à fait possible. Il consiste à combiner les deux types d'enrichissement et permet donc d'ajouter à la requête à la fois les propriétés spécifiques (les sommets du sous arbre de racine a) et celles générales (les sommets du chemin de a à r). L'ensemble de propriétés de la requête enrichie sera dans ce cas à partir d'une propriété a :

$$(B_Q \cup V' \cup \{v_i,\ i=2..n\ /\ p=(v_1,v_2,\ \ldots\ ,v_n)\ \textit{est le chemin de}\ a=v_1\ \textit{à r racine de T}\}) \cap P$$

où V' est l'ensemble des sommets du sous arbre T' de T enraciné en a.

3.2.3 Généralisation à l'utilisation de plusieurs ontologies de domaine

Parmi l'ensemble des propriétés choisies pour caractériser (indexer) les sources biologiques étudiées, plusieurs sous-ensembles peuvent être organisées en hiérarchies constituant des ontologies de domaines notamment les propriétés relatives aux fonctions moléculaires des protéines et celles relatives à la fréquence de mise à jour et à la couverture des sources. De fait, la prise en compte de l'ontologie des organismes modèles (*section 3.2, figure 3.1*) peut être combinée avec ces autres ontologies.

Fonctions Moléculaires

Les fonctions moléculaires [8] [9] des protéines sont nombreuses et diverses et la représentation de chaque fonction par une propriété dans le contexte formel entraînera une explosion inutile de celui-ci. Pour éviter ce problème, nous n'avons retenu dans l'ontologie que les fonctions les

[8] http ://www.geneontology.org/
[9] http ://www.godatabase.org/cgi-bin/amigo/go.cgi

plus réputées et qui sont susceptibles de donner une annotation efficace sur les sources. Parmi les classes de propriétés (nœuds proches de la racine de l'ontologie) figurant dans l'ontologie on distingue la famille des molécules ayant la propriété de liaison (*Binding*). Celle-ci comprend les molécules qui se lient spécifiquement aux acides aminés (*Amino Acid Binding*), celles se liant spécifiquement aux antigènes (*Antigen Binding*) et celles se liant spécifiquement aux récepteurs (*Receptor Binding*). Chacune des trois propriétés citées regroupe à son tour un ensemble de propriétés plus spécifiques. Étant donné cette hiérarchie de propriétés, il est possible d'effectuer les enrichissement définis dans la section 3.2.

Couverture et fréquence de mise à jour des sources

La classification des propriétés (valeurs possibles) des champs relatifs à la fréquence de mise à jour, à la date de la dernière modification et à la couverture des sources ne donne pas une hiérarchie arborescente comme celles des propriétés relatives aux organismes ou aux fonctions moléculaires. En revanche, il s'agit simplement d'ordonner les valeurs possibles sur un même axe permettant d'effectuer les deux parcours de généralisation et spécialisation qui consistent simplement à parcourir l'axe dans un sens ou dans l'autre selon le type de d'enrichissement choisi. Cet ordonnancement doit permettre, par exemple, suite à l'enrichissement par généralisation d'une requête demandant les sources ayant une fréquence de mise à jour mensuelle de ramener aussi celles ayant une fréquence trimestrielle, semestrielle, ou annuelle.

3.3 Apports de l'enrichissement sémantique de la requête

Après l'enrichissement de la requête, on effectue les mêmes étapes que celles décrites dans la section 2 à savoir :

- Insérer la requête enrichie dans le treillis de Galois qui représente la hiérarchie des sources.
- Localiser, dans le treillis, le concept ayant la plus petite intension contenant les propriétés de la requête. Soit C_{RE}=*(*A_{RE},B_{RE}*)* ce concept.
- Insérer dans le résultat les éléments de $R_0 = A_{RE} \backslash \{Query\}$ si cet ensemble n'est pas vide.
- Récupérer les parents directs (les subsumants les plus spécifiques) de C_{RE}. Soit $\mathcal{C}_1$ l'ensemble de ces concepts. Pour chaque concept $C_{\mathcal{C}_{1i}} = (A_{\mathcal{C}_{1i}}, B_{\mathcal{C}_{1i}})$ de $\mathcal{C}_1$, on construit l'ensemble des réponses de rang 1, R_1. $R_1 = \bigcup_{i=1..|\mathcal{C}_1|} \{(A_{\mathcal{C}_{1i}} \backslash R_0)$ pour tout $A_{\mathcal{C}_{1i}}$ intension de concept $C_{\mathcal{C}_{1i}} = (A_{\mathcal{C}_{1i}}, B_{\mathcal{C}_{1i}})$ tel que $B_{\mathcal{C}_{1i}} \neq \emptyset\}$. $|\mathcal{C}_1|$ étant le cardinal de $\mathcal{C}_1$.
- Répéter récursivement l'étape précédente jusqu'au concept le plus général $C_G = (A_G, B_G)$ ayant une intension non vide B_G=$\emptyset$.

La proposition suivante met en valeur l'apport de l'enrichissement de la requête qui sera prouvé formellement par la suite.

Proposition 3.2 *L'enrichissement de la requête permet d'avoir :*

1. *Un résultat plus riche en termes de nombre de sources pertinentes.*
2. *Un ordre plus precis sur les sources contenues dans le résultat.*

Preuve 3.2 *Soit* C_Q=*(*A_Q,B_Q*)* *le concept de la requête initiale avec* A_Q=*{Query} et* B_Q *est l'ensemble des propriétés souhaitées par l'utilisateur. On notera* C_{QE}=*(*A_Q,B_{QE}*)* *le concept de la requête enrichie où* B_{QE}=$B_Q \cup P_A$. P_A *est l'ensemble de propriétés apportées par l'enrichissement sémantique de la requête.*

- P_A=($B_Q \cup \{v_i$, i=2..n / p=(v_1,v_2, ... ,v_n) est le chemin de x=v_1 à r racine de T$\}$)$\cap$P *dans le cas d'un enrichissement par généralisation (section 3.2.1).*

- P_A=(B_Q∪V')∩P *dans le cas de l'enrichissement par spécialisation (section 3.2.2).*
- P_A=(B_Q∪V'∪{v_i, i=2..n / p=(v_1,v_2, ... ,v_n) est le chemin de x=v_1 à r racine de T})∩P *dans le cas d'un enrichissement mixte (section 3.2.2).*

Soit C_R=(A_R,B_R) le concept du treillis modifié suite à l'insertion d'une requête simple (après l'insertion de C_Q) et soit C_{RE}=(A_{RE},B_{RE}) celui du treillis modifié à l'insertion d'une requête enrichie (après insertion de C_{QE}).

1. Par définition du concept de la requête enrichie on a B_{RE}=B_R∪P_A ce qui implique que B_R⊆B_{RE}. Et comme C_R et C_{RE} sont deux concepts de Galois (appartiennent au treillis de Galois) l'inclusion précédente est équivalente à la relation de subsomption C_{RE}≼C_R. Ainsi C_{RE} est plus spécifique que C_R et par suite si on considère que le treillis de Galois est formé par un ensemble de niveaux entre les deux concepts ⊤ (top) et ⊥ (bottom), le concept C_{RE} sera dans un niveau plus bas que celui de C_R. Être à un niveau plus bas dans la hiérarchie de concept implique forcément avoir plus de concepts parents dans le treillis et, en conséquence, avoir plus de sources dans le résultat qui n'est autre que l'union des extensions des concepts parents de C_{RE} le concept à intension vide exclus.

En conclusion, la réponse à une requête enrichie est plus riche en nombre de sources.

2. On a montré précédemment que C_{RE}≼C_R.

Ceci implique que A_{RE}⊆A_R et donc A_{RE}\{Query} ⊆ A_R\{Query}. Ainsi le nombre de sources ajoutées à cette étape au résultat d'une requête enrichie est inférieur à celui ajouté au résultat d'une requête normale. De ce fait le nombre de sources qui auront le numéro de cette étape comme rang dans le résultat final d'une requête enrichie sont moins nombreux. Le reste des sources qui avaient ce même rang dans le résultat de la requête simple auront un rang plus grand et le changement sur les rangs se fait en cascade jusqu'aux dernières sources ajoutées.

En conclusion, l'enrichissement de la requête permet, en plus de l'enrichissement du résultat en nombre de sources, de réordonner plus précisément les sources selon leur pertinence et séparer éventuellement des sources qui avaient le même degré de pertinence (rang) dans le cas de la requête simple.

Pertinence des sources ajoutées au résultat

La pertinence des sources ajoutées découle du fait que celles ci figurent dans le résultat grâce à une ou plusieurs propriétés ajoutées à la requête lors de l'enrichissement sémantique. Or, comme détaillé dans la section 3.2, tout ajout de propriété à la requête est dicté par une relation sémantique traduisant une certaine similarité entre les contenus des sources ayant la propriété ajoutée. De ce fait, le risque d'apparition de sources non pertinentes dans la réponse à retourner à l'utilisateur est écarté. Toutefois le degré de pertinence des sources ajoutées peut être faible. En effet si le chemin, dans T, de la propriété initiale a à une propriété ajoutée lors de l'enrichissement est assez long et que cette propriété permet d'avoir une source supplémentaire dans le résultat alors deux cas se présentent. S'il s'agit d'un enrichissement par généralisation, la source ajoutée peut être très générale et par suite peu précise par rapport à ce qui est souhaité par l'utilisateur. Et s'il s'agit d'un parcours par spécialisation, la source ajoutée peut couvrir avec beaucoup de détails seulement une petite partie de ce que l'utilisateur demande. Mais en aucun cas une source ajoutée ne peut être totalement écartée par rapport à ce qui est souhaité par l'utilisateur.

Illustration

Considérons une requête Q ayant les propriétés *Eucaryotes* et *Annotation Gene Ontology*. La réponse à cette requête est formée des sources *S1, S2, S3* et *S7* retournées uniquement en raison du partage de la propriété *Annotation Gene Ontology*. Rencontrées toutes à la même étape, ces sources ont le même degré de pertinence (même rang : 1) par rapport à la requête. Examinons comment l'enrichissement sémantique de la requête (ici par généralisation) permet non seulement d'avoir plus de précision sur l'ordre de ces mêmes sources, mais aussi d'ajouter d'autre sources au résultat.

La requête enrichie par spécialisation, Q_E, correspondante à Q a les propriétés *Annotation Gene Ontology, Animaux, Vertébrés, Homme* et *Souris* (*voir section 3.2.2 pour plus de détails*). Les modifications apportées au treillis après l'insertion de C_Q (concept de la requête enrichie) sont données par la figure 3.2. L'ensemble des sources formant la réponse à Q_E dans l'ordre décroissant de la pertinence par rapport à la requête est le suivant :

1. S3 : *TIGR-HGI* qui partage avec la requête enrichie les deux propriétés *Homme* et *Annotation Gene Ontology.*
1. S7 : *Mouse Genome DB* qui partage avec la requête enrichie les deux propriétés *Souris* et *Annotation Gene Ontology.*
1. S6 : *ENSEMBL* qui partage avec la requête enrichie la propriété *Animaux.*
1. S5 : *HUGE* qui partage avec la requête enrichie la propriété *Homme.*
1. S8 : *Vega Genome Browser* qui partage avec la requête enrichie la propriété *Vertébrés.*
2. S1 : *Swiss Prot* qui partage avec la requête enrichie la propriété *Annotation Gene Ontology.*
2. S2 : *RefSeq* qui partage avec la requête enrichie la propriété *Annotation Gene Ontology.*

Dans le résultat, réponse à Q_E, on remarque bien les deux points cités dans la proposition. En effet, le nombre de sources constituant le résultat est passé de quatre à sept (on a en plus des sources figurant dans le premier résultat (S1, S2, S3 et S7), les sources S5, S6 et S8) et le rang des sources du premier résultat n'est plus le même : S1 et S2 ne sont plus dans le premier rang.

L'exemple présenté illustre l'apport de l'enrichissement sémantique par spécialisation et le raisonnement est le même pour l'enrichissement sémantique par généralisation et l'enrichissement mixte. En effet si on considère la requête enrichie par généralisation, détaillée dans la section 3.2.1 et ayant les propriétés *Tout Organisme* et *Annotation Gene Ontology*, le résultat sera :

1. S1 : *Swiss Prot* qui partage avec la requête enrichie les deux propriétés *Tout Organisme* et *Annotation Gene Ontology.*
1. S2 : *RefSeq* qui partage avec la requête enrichie les deux propriétés *Tout Organisme* et *Annotation Gene Ontology.*
2. S3 : *TIGR-HGI* qui partage avec la requête enrichie la propriété *Annotation Gene Ontology.*
2. S7 : *Mouse Genome DB* qui partage avec la requête enrichie la propriétés *Annotation Gene Ontology.*
2. S4 : *GPCRDB* qui partage avec la requête enrichie la propriété *Tout Organisme.*

L'enrichissement mixte permet d'avoir l'union des deux résultats avec des changements éventuels sur l'ordre des sources.

Il est à noter ici qu'une réflexion est à poursuivre, grâce notamment à des tests intensifs du système, pour déterminer les cas où il est préférable d'enrichir par généralisation et ceux où il est préférable d'enrichir par spécialisation.

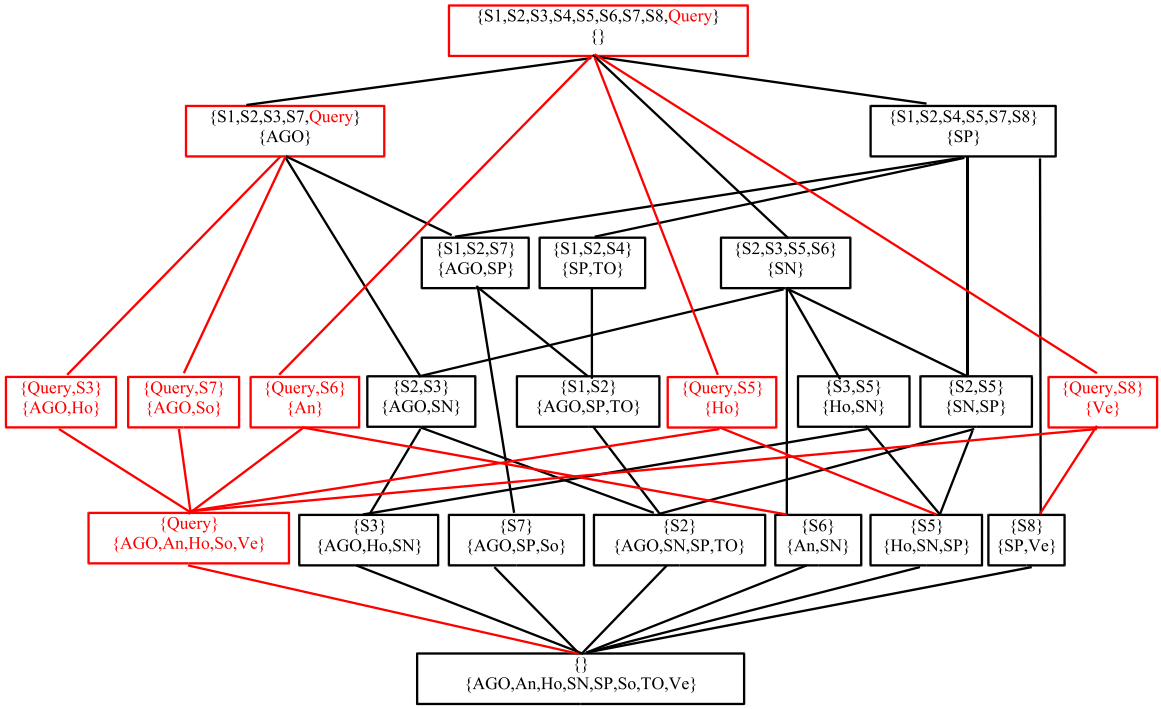

FIG. 3.2 – **Treillis Θ(K) (*cf figure 2.1*) modifié suite à l'insertion d'une requête enrichie par spécialisation**

3.4 Positionnement par rapport aux travaux voisins

L'idée de départ de notre travail était de permettre aux utilisateurs notamment les biologistes de trouver des banques de données biologiques accessibles sur le web et susceptibles de contenir des informations relatives aux besoins exprimés dans leurs requêtes. Or ces banques de données, étant mises en place par différents organismes, ont des schémas et par suite des modes d'interrogation différents. Cette hétérogénéité ramène notre problématique à celle d'une médiation entre les utilisateurs et les banques de données biologiques. À la différence des systèmes de médiation [BFG+02], notre travail se limite à la recherche des sources (banques de données) pour ramener aux utilisateurs les plus pertinentes vis-à-vis des informations exprimées dans les requêtes. La phase d'interrogation de ces sources n'est pas traitée par le système et repose sur une interaction ultérieur de l'utilisateur avec les interfaces offertes par les sources elle-mêmes.

La décision portant sur la pertinence des sources par rapport à une requête donnée suppose la connaissance d'un ensemble de critères qu'elles vérifient. Ceci nous a amené à penser à une analogie entre notre problématique et la recherche documentaire. Or les propositions les plus récentes visent à baser la recherche documentaire sur des formalismes à fondements mathématiques, notamment les treillis de Galois [GMM95b, CR00], leur permettant d'avoir des résultats exacts et prouvables. La différence de notre travail par rapport à ces propositions réside dans le fait que ceux-ci passent par des moteurs de recherche existants (*Google, Yahoo ...*) pour avoir un premier filtrage des documents leur permettant de ne garder qu'une collection relativement réduite qui sera par la suite classée dans un treillis de Galois en se basant sur les termes d'indexation (fournis automatiquement par les moteurs de recherche) des documents de cette collection [CR00]. L'étape suivante consiste à classer la requête initiale dans le treillis pour déterminer un

nouveau classement des éléments de la collection par rapport à cette requête. Étant donné que l'utilisation des moteurs de recherche existants ne permet pas de localiser les banques de données souvent cachées derrière des interfaces d'interrogation, nous avons adopté notre propre solution consistant à recueillir auprès d'experts du domaine un ensemble d'informations (métadonnées) nécessaires pour la caractérisation de chaque banque dans un catalogue. Celui ci servira de base pour la création du treillis de Galois structurant la collection et permettant de retourner, pour chaque requête, l'ensemble des sources pertinentes.

À ce niveau, la méthode proposée permet de donner des réponses exactes aux requêtes qui lui sont passées. Toutefois, faute d'informations explicites sur la sémantique des informations disponibles dans le catalogue, la méthode peut passer sous silence de sources parfois plus pertinentes que celles retournées. Pour résoudre ce problème nous avons incorporé les propriétés ayant des relations sémantiques dans des ontologies de domaines. L'utilisation de ces ontologies se fait dans la phase d'enrichissement des requêtes avant de les insérer dans le treillis. L'enrichissement sémantique de la requête constitue l'originalité de notre proposition. En effet les travaux de réécriture de requêtes ont généralement pour but de permettre l'interrogation de plusieurs sources hétérogènes, d'optimiser le traitement et/ou d'éviter les conflits qu'une requête peut poser [BFG+02, Hal01]. Ils consistent à obtenir une requête équivalente par réécriture de la requête initiale en utilisant soit un language propre à l'application [BFG+02] soit un ensemble de termes définis au préalable [Hal01] (par exemple les vues dans les bases de données évoquées dans la section 1.4).

3.5 Réalisation

L'implémentation effectuée dans le cadre de ce travail concerne uniquement la méthode de classification et de recherche de sources génomiques détaillée dans le deuxième chapitre.

Il s'agit d'un programme JAVA qui prend en entrée une requête sous forme d'un ensemble de propriétés et génère en sortie la liste ordonnée des sources pertinentes correspondant à cette requête. Un petit descriptif est joint à chaque source ainsi qu'un ensemble d'informations permettant aux utilisateurs d'y accéder. Parmi ces information on trouve le nom de la source et ses différentes URLs disponibles. Les informations sur les sources sont récupérées des objets instances de la classe *BioSource*. Ces instances sont crées à partir d'un fichier XML *sources.xml* qui représente la collection des sources disponibles avec leurs propriétés (*cf. tableau 2.1.* La création de ces instances est assurée par la classe *CreationSource* qui parcourt les balises du fichier *sources.xml* pour en extraire les attributs de la classe *BioSource*.

Pour la construction du treillis de Galois, nous utilisons la classe *CreationTreillisAPartirFichier*, implémentation de l'algorithme de construction incrémentale de treillis de Galois [GMM95a] par Benoît Vallayer dans l'équipe ORPAILLEUR. Cette classe prend en entrée un deuxième fichier XML *concept.xml* représentant le contexte formel et génère le treillis de Galois correspondant sous forme d'une instance de la classe *Treillis* qui fait le lien entre un ensemble d'objets de type *ConceptGalois* et un ensemble d'objets de type *Lien*.

La classe *SourceSelection* analyse la requête de l'utilisateur pour en extraire des propriétés contenues dans le contexte. À partir de ces propriétés elle construit le concept (une instance de *ConceptGalois*) correspondant à la requête qui sera inséré dans le treillis de Galois. Cette insertion est effectuée à l'aide de la méthode *constructionIncrementaleTreillisGalois* de la classe *Treillis* qui prend en paramètre le concept construit et modifie l'objet *Treillis* sur lequel la méthode a été appelée.

La recherche des sources pertinentes pour la requête est assurée par la méthode *determi-*

nerVoisinage de la classe *SourceSelection*. Elle retourne un vecteur de noms de sources qui sera passé en paramètre à la méthode *listerSources* de la même classe. Celle-ci localisera par la suite les objets *BioSources* à partir du vecteur des noms en utilisant la méthode *getSourceByName*. L'ensemble des sources sera enfin retourné à la classe initiale *Test_sources* qui constitue le point d'entrée du programme et fait les appels aux autres classes en leur spécifiant les fichiers XML à considérer ainsi que les requêtes posées. Cette classe s'occupe de l'affichage de la liste de sources pertinentes pour la requête en entrée à partir de l'ensemble d'objets *BioSources*.

La réalisation brièvement présentée est un prototype qui nous a permis de tester la méthode de classification de recherche de sources génomiques en utilisant les treillis de Galois. Nous envisageons de perfectionner ce prototype dans un premier temps puis d'y inclure l'enrichissement sémantique de requêtes avant de le mettre à la disposition des utilisateurs avec un ensemble de sources plus complet en vue de tests. Des améliorations resteront envisageables en fonction de la satisfaction des utilisateurs suite à ces tests.

Conclusion

La proposition présentée dans ce chapitre consiste à exploiter les relations sémantiques existant entre les propriétés des sources exprimées dans une ou plusieurs ontologies de domaine. La prise en compte de ces relations permet d'enrichir la requête initiale d'un utilisateur par de nouveaux termes (propriétés) dans le but d'avoir le résultat le plus riche possible tout en respectant l'intérêt de l'utilisateur (exprimé dans sa requête initiale). On distingue essentiellement deux types d'enrichissement sémantiques : par spécialisation et par généralisation et la combinaison des deux est envisageable.

L'enrichissement sémantique de la requête nous a permis d'avoir non seulement plus de sources dans le résultat mais aussi un ordre plus pertinent sur ces sources.

Conclusion et perspectives

Dans ce travail nous avons étudié le problème de la classification et de l'identification des ressources génomiques pertinentes pour la réponse à une question donnée. Nous avons eu recours à des spécialistes du domaine biologique pour la collecte d'informations disponibles sur les métadonnées associées à ces ressources et concernant leur qualité aussi bien que leur contenu. Ces informations nous ont permis de classer les sources dans une hiérarchie (treillis de Galois) facilitant la récupération de l'ensemble des sources pertinentes classées dans l'ordre décroissant de leur pertinence pour chaque requête posée par les utilisateurs.

L'amélioration de la méthode de recherche des sources dans une hiérarchie consistant à l'enrichissement sémantique des requêtes à partir des ontologies de domaine nous a permis d'enrichir les résultats à rendre aux utilisateurs tout en restant dans le cadre de leurs préférences.

La méthode proposée s'appuie sur le fondement mathématique des treillis de Galois ce qui nous à permis de prouver l'exactitude de ses résultats. Ce fondement mathématique permettra d'explorer encore plus la piste de recherche d'informations non seulement dans le domaine biologique mais aussi dans tout autre domaine d'application tout en prouvant à chaque étape les résultats obtenus.

La prise en compte de la sémantique valorise encore plus notre proposition vu l'intérêt grandissant accordée à cet aspect dans la majorité des domaines de recherche notamment celui du web sémantique et des services web.

Ce travail laisse envisager plusieurs perspectives qui s'avèrent intéressantes.

Il serait d'abord intéressant de finaliser ce travail sur un annuaire plus conséquent référençant un plus grand nombre de sources biologiques accessibles sur le web, annotées par un ensemble plus vaste de propriétés. Une expérimentation sur cet annuaire permettrait en particulier d'évaluer le degré de satisfaction de l'utilisateur biologiste et de définir des heuristiques permettant de décider quand un enrichissement de la requête par généralisation ou par spécialisation est souhaitable.

Le maintien de la cohérence de l'annuaire nécessite une mise à jour fréquente des métadonnées associées aux sources. Ainsi il serait intéressant d'automatiser cette tâche, en appliquant un processus de fouille sur les sites web des sources, puis en effectuant les modifications dans la collection en cas de mises à jour. En plus du maintien à jour, le processus de fouille permettrait aussi de repérer de nouvelles sources et d'extraire de nouvelles métadonnées pour un enrichissement de l'annuaire.

Une autre perspective concernant le traitement de requêtes consiste à effectuer une pondération de leurs propriétés. Cette pondération permettra de développer encore plus la prise en compte de la sémantique à la fois lors du traitement et de l'enrichissement des requêtes.

Enfin il sera utile de passer à l'invocation des sources pour répondre directement aux requêtes posées et ceci permettra d'aboutir à la construction d'un médiateur.

Bibliographie

[BFG+02] Alain Bidault, Christine Froidevaux, Hélène Gagliardi, François Goasdoué, Chantal Reynaud, Marie-Christine Rousset, and Brigitte Safar. Construction de médiateurs pour intégrer des sources d'information multiples et hétérogènes : le Projet PICSEL. *Revue I3 : Information - Interaction - Intelligence*, 2(1) :9–59, 2002.

[BHS02] Bettina Berendt, Andreas Hotho, and Gerd Stumme. Towards semantic web mining. In *Proceedings of the First International Semantic Web Conference on The Semantic Web*, pages 264–278. Springer-Verlag, 2002.

[Bir67] Garrett Birkhoff. *Lattice Theory*, volume 25 of *ASM Colloquium Publications*. ams, Providence, RI, 3rd edition, 1967. 1st ed., 1940 ; 2nd ed., 1948.

[BLHL01] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5) :35–43, mai 2001.

[BM70] M. Barbut and B. Monjardet. *Ordre et classification : Algèbre et combinatoire (2 tomes)*. Hachette, Paris, 1970.

[CBT03] Jean Charlet, Bruno Bachimont, and Raphael Troncy. Ontologies pour le web sémantique. In Jean Charlet, Philippe Laublet, and Chantal Reynaud, editors, *Action spécifique 32 CNRS/STIC Web sémantique Rapport final*, volume 2, pages 43–63, octobre 2003.

[CG04] Vishnu Challam and Susan Gauch. Contextual information retrieval using ontology-based user profiles. *Nineteenth National Conference on Artificial Intelligence (AAAI-04), San Jose, CA*, Juillet 2004.

[CLR92] Thomas Cormen, Charles Leiserson, and Ronald Rivest. *Introduction à l'algorithmique*. Dunod, 1992.

[CR93] Claudio Carpineto and Giovanni Romano. Galois : An order-theoretic approach to conceptual clustering. *Proceedings of 10th International Conference on Machine Learning, Amherst*, pages 33–40, June 1993.

[CR00] Claudio Carpineto and Giovanni Romano. Order-theoretical ranking. *Journal of the American Society for Information Science*, 51(7) :587–601, 2000.

[Gal04] Michael Y. Galperin. The molecular biology database collection : 2004 update. In National Center for Biotechnology Information, National Library of Medicine, and National Institutes of Health, editors, *Nucleic Acids Research*, volume 32 Database issue D3-D22, Bethesda, USA, 2004.

[GL02] Michael Gruninger and Jintae Lee. Ontology : applications and design. *Communications of the ACM*, 45(2) :39–41, feb 2002.

[GM93] A. Guenoche and I. Van Mechelen. Galois approach to the induction of concepts. In I. Van Mechelen, J. Hampton, R.S. Michalski, and P. Theuns, editors, *Categories and*

Concepts. Theoretical Views and Inductive Data Analysis, pages 287–308. Academic Press, London, 1993.

[GMM95a] R. Godin, G. Mineau, and R. Missaoui. Incremental structuring of knowledge bases. In G. Ellis, R. A. Levinson, A. Fall, and V. Dahl, editors, *Proceedings of the 1st International Symposium on Knowledge Retrieval, Use, and Storage for Efficiency (KRUSE'95), Santa Cruz (CA), USA*, pages 179–193. Department of Computer Science, University of California at Santa Cruz, 1995.

[GMM95b] R. Godin, G. Mineau, and R. Missaoui. Méthodes de classification conceptuelle basées sur les treillis de galois et applications. *Revue d'intelligence artificielle*, 9(2) :105–137, 1995.

[Gru93] T.R. Gruber. A translation approach to portable ontology specification. *Knowledge Acquisition*, 5(2) :199–220, 1993.

[Gue90] Alain Guenoche. Construction du treillis de galois d'une relation binaire. *Mathmatiques, Informatique et Sciences Humaines*, 28ème année(109) :41–53, 1990.

[Hal01] Alon Y. Halevy. Answering queries using views : A survey. *The VLDB Journal*, 10(4) :270–294, 2001.

[KB00] Raymond Kosala and Hendrik Blockeel. Web mining research : a survey. *SIGKDD Exploration Newsletter*, 2(1) :1–15, 2000.

[MS01] Alexander Maedche and Steffen Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16(2) :72–79, 2001.

[Pie02] Gay Pierra. Un modèle formel d'ontologie pour l'ingénierie, le commerce électronique et le web sémantique : Le modèle de dictionnaire sémantique plib. In *Journées Scientifiques WEB SEMANTIQUE*, Paris, Octobre 2002.

[PRSV02] Nathalie Pernelle, Marie-Christine Rousset, Henri Soldano, and Veronique Ventos. ZooM : a nested galois lattices-based system for conceptual clustering. *Journal of Experimental and Theoritical Artificial Intelligence (JETAI)*, 14(2) :157–187, september 2002.

[SQ02] José Saias and Paulo Quaresma. Semantic enrichment of a web legal information retrieval system. In T. Bench-Capon, A. Daskalopulu, and R. Winkels, editors, *Frontiers in AI and Applications*, volume 89, pages 11–20, London UK, Décembre 2002. JURIX'2002, Fifteenth Annual International Conference on Legal Knowledge and Information Systems.

[SQ03] José Saias and Paulo Quaresma. A methodology to create ontology-based information retrieval systems. In Fernando Moura-Pires and Salvador Abreu, editors, *Progress in Artificial Intelligence, 11th Portuguese Conference on Artificial Intelligence, EPIA, 2003*, volume 2902 of Lecture Notes in Computer Science, pages 424–434, Beja Portugal, Décembre 2003.

[VM01] Petko Valtchev and Rokia Missaoui. Building concept (galois) lattices from parts : Generalizing the incremental methods. In *Proceedings of the 9th International Conference on Conceptual Structures*, pages 290–303. Springer-Verlag, 2001.