Numéro d'Identification

THÈSE

Université de Montpellier II
Sciences et Techniques du Languedoc

Discipline Informatique
Formation Doctorale Informatique
Ecole Doctorale Information, Structures, Systèmes

Cartographie des connaissances : l'intégration et la visualisation au service de la biologie

Application à l'ingénierie des connaissances et à l'analyse de données d'expression de gènes

Fabien Jalabert

Soutenue le 5 décembre 2007 pour obtenir le grade de docteur de l'Université Montpellier 2 devant le jury composé de :

Rapporteurs :	Christine Froidevaux	Professeur, Université Paris Sud (Paris 11)
	Guy Perrière	Directeur de Recherche CNRS, Université Claude Bernard - Lyon 1
Examinateurs :	Yvon Cayre	Professeur, Université Pierre et Marie Curie - Paris 6
	Michel Crampes	Enseignant chercheur (HDR), Ecole des Mines d'Alès (directeur de thèse)
	Vincent Derozier	Enseignant chercheur, Ecole des Mines d'Alès (encadrant de proximité)
	Thérèse Libourel	Professeur, Université Montpellier II
	Sylvie Ranwez	Enseignant chercheur, Ecole des Mines d'Alès (encadrant de proximité)
	Christophe Roche	Professeur, Université de Haute Savoie
Invitée :	Isabelle Mougenot	Maître de conférences, Université Montpellier II

à Stéphanie et Clément

Remerciements

Je souhaite remercier ici …

La direction de ma thèse. Michel Crampes m'a offert sa confiance en me laissant la liberté de choisir mes directions de recherche. Il m'a alors appris à prendre du recul et à conserver à l'esprit les attentes concrètes de l'utilisateur. Sylvie Ranwez m'a encadré tout au long de ces années. Elle a partagé avec moi son expertise du domaine de l'ingénierie des connaissances et son expérience en matière de communication. Vincent Derozier a lui aussi participé à mon encadrement de proximité. Il a partagé sa passion des sciences du vivant et m'a soutenu moralement dans les périodes difficiles. Je les remercie tous deux pour la patience et la compréhension dont ils ont fait preuve quotidiennement.

Isabelle Mougenot qui s'est investie dans le suivi de ma thèse. Elle m'a fait bénéficier de son expérience et de son recul en bioinformatique et m'a permis de m'insérer dans la communauté.

Les membres du jury pour l'intérêt qu'ils ont montré à mes travaux. Christine Froidevaux et Guy Perrière ont accepté de rapporter mon mémoire. Yvon Cayre, Thérèse Libourel et Christophe Roche ont accepté le rôle d'examinateur.

La direction de l'école des mines d'Alès et du LGI2P pour m'avoir accueilli dans leur établissement. Je remercie en particulier Pascal Poncelet qui s'est particulièrement impliqué dans le suivi de ma thèse.

Mes amis doctorants au LGI2P et à l'université de Montpellier : Frédéric Souchon puis Nicolas Desnos avec qui nous avons partagé le quotidien d'un bureau, et Jean Villerd, mon homologue dans l'équipe du LGI2P. Leur soutien moral, leur écoute, et leurs points de vue ont contribué au bon déroulement de ces quatre années. Je conserve un agréable souvenir des multiples discussions que nous avons eu avec Didier Schwab, Rheena Shetty, Olivier Gout et Saber Aloui.

Les chercheurs et ingénieurs du LGI2P qui m'ont conseillé et sontenu : Sylvain Vauttier, Christelle Urtado, Gérard Dray, Jacky Montmain, Michel Plantié, Thomas Lambolais, Stefan Janaqi, Vincent Chapurlat, Annie Liothin, François Trousset et François Pfister, Pierre Runtz.

Les membres du centre de documentation, Françoise Armand et Véronique qui m'ont accompagné et formé dans une tâche jusque là éludée par mon cursus.

Sylvie Cruvellier qui m'a évité de me perdre dans les méandres de l'administration.

Jean-Christophe Lallement-Soboul et son équipe de l'école de l'ADN qui réalisent un merveilleux travail et qui m'on formé bénévolement à la biologie moléculaire.

Vincent Ranwez pour m'avoir conseillé et aidé à découvrir une nouvelle communauté.

Jacques Bourguignon et son équipe pour m'avoir accueilli au sein de son laboratoire.

Il est difficile de résumer quatre années de sympathie en quelques lignes. Et j'adresse toutes mes excuses aux oubliés de cette page.

Je ne saurais terminer ces remerciements sans citer ma famille et mes parents. Cette thèse n'aurait pas abouti sans les encouragements et la tendresse que m'ont offerts Stéphanie et Clément. Je les remercie pour leur compréhension ; partager les turpitudes d'un doctorant n'est pas chose facile.

Sommaire général

Sommaire détaillé

Introduction

« La dernière chose qu'on trouve en faisant un ouvrage est de savoir celle qu'il faut mettre la première »

BLAISE PASCAL, PENSEES SUR L'ESPRIT ET LE STYLE, 1670

Ce mémoire s'inscrit dans un axe stratégique du groupement des Ecoles des Mines intitulé GEMBIO. Dans ce programme, deux collaborations ont été initiées avec des laboratoires de biologie travaillant sur des thématiques différentes. Le premier, qui appartient à l'Institut Pasteur de Paris, étudie le parasite *Plasmodium Falciparum* responsable du paludisme (*malaria* chez les anglophones). Cette affection tropicale, la plus répandue dans le monde touche près de 40% de la population des pays les plus pauvres, essentiellement en Afrique. Première cause de mortalité chez les enfants de moins de 5 ans, elle est responsable de plus d'un million de décès chaque année. Les problèmes bioinformatiques de ces chercheurs concernent l'analyse de données d'expression issues de puces à ADN : ils quantifient les ARN messagers présents dans la cellule à un instant donné et dans certaines conditions expérimentales. Dans le cas de *Plasmodium Falciparum*, le but est de mesurer la cinétique des gènes transcrits sous la pression d'une drogue afin d'évaluer l'impact de cette drogue jusqu'à la mort du parasite. En utilisant la similarité d'expression, le chercheur tente de prédire des caractéristiques de gènes (fonction, localisation, processus biologique, etc.). Les puces à ADN permettent une approche globale à partir de laquelle on met en évidence des éléments d'intérêt qu'il faut par la suite étudier avec plus de précision (RT-PCR, etc.). Elles peuvent s'appliquer à des génomes entiers (ici *Plasmodium Falciparum* comporte près de 5300 gènes, on estime que l'homme en possède près de 30 000).

La seconde collaboration à l'origine de nos travaux s'effectue dans le cadre d'un programme transversal ToxNuc-E. Ce programme réunit actuellement plus de 600 chercheurs en physique, chimie, biologie, etc. autour d'une thématique commune : la toxicologie nucléaire environnementale. Différents organismes en sont partenaires : CEA, CNRS, INRA, INSERM et MRNT. ToxNuc-e est organisé en 15 sous-projets. En ce qui nous concerne, des discussions ont été menées avec les coordonnateurs de deux projets différents. Jacques Bourguignon travaille avec son équipe au sein du projet Arabidopsis[1]. Dans une approche protéomique, ils mesurent l'impact des radiations provoquées par certains métaux lourds sur les cellules de ce végétal. Les données correspondent donc à la quantité de certaines protéines et certains métabolites présents dans la cellule. Le principal problème bioinformatique évoqué est la difficulté de croiser l'information partagée et dispersée. Des problèmes similaires liés à la recherche d'information et la veille scientifique et technologique sont aussi évoqués par Sylvie Chevillard co-coordinatrice du projet Nephrotoxicité et Toxico-cancérogénèse. Elle aussi suit une approche transcriptomique.

D'une façon générale, nos contacts avec des chercheurs en biologie font ressortir des thématiques diverses de traitement de l'information : analyse des données (fouille, croisement, échange et synthèse au sein de plusieurs outils), recherche d'information (bibliographie,

[1] Du nom d'un organisme modèle en biologie végétale, l'*Arabidopsis Thaliana*, connue aussi sous le nom Arabette des Dames.

nombreux portails en ligne), problème de définition du vocabulaire et plus généralement problèmes liés à la langue, etc. D'un point de vue informatique, ces problèmes s'inscrivent dans plusieurs axes : intégration de données, visualisation de l'information, ingénierie des connaissances, analyse de données, etc. Mais rapidement, une phrase récurrente revient en de multiples occasions dans le dialogue : *« j'ai régulièrement plus d'une dizaine de fenêtres ouvertes sur des bases de données spécifiques, et pour croiser les informations contenues, je suis perdu … »*. Les données biologiques, massives, hétérogènes et distribuées, doivent être synthétisées et visualisées sous une forme utile et efficace, dans le contexte précis d'un besoin et d'une tâche pour l'utilisateur. L'outil qui veut répondre à ce besoin doit donc permettre de filtrer et contextualiser l'information. Il doit être suffisamment souple pour s'intégrer dans de nombreux contextes applicatifs, et doit enfin permettre d'accéder à une information disparate tant dans sa forme, que dans le contenu, la localisation, etc.

Nous avons identifié un réel besoin de « **cartographie de la connaissance biologique »**, un terme qui regroupe différentes notions[1]. *« **Cartographie** »* donne une dimension spatiale et visuelle ; la carte se veut une représentation intuitive, utile et utilisable, partageable, permettant de se repérer dans un espace complexe que l'on souhaite percevoir sous un nouvel angle, à différentes échelles. La cartographie est ainsi une invitation au voyage, à aller voir *« ce qu'il y a derrière la colline »* [Buttenfield and Weber 1994]. Elle s'accorde donc bien avec l'utilisation qui en est faite d'analyse exploratoire de données. Enfin, ce terme fait référence à un objet du quotidien dont personne ne remettrait en cause l'utilité. La seconde notion importante réside dans le terme *« **la connaissance biologique** »*. Nous proposons d'intégrer dans cet outil visuel la connaissance du vivant dans sa globalité, sans restriction *a priori*. Elle peut être liée à un contexte plus ou moins général, contenue dans des articles de recherche, des annotations des portails du domaine ou des modèles plus ou moins formels.

Concrètement, à cette nouvelle approche de l'intégration de données hétérogènes, orientée utilisateur, est associé un prototype appelé I^{2}DEE[2]. Cet environnement est constitué de deux composantes principales.

A un niveau serveur, un entrepôt de données est construit. Il est structuré autour d'un graphe typé et valué qui offre des avantages en termes de souplesse, performance, extensibilité, et rapidité d'ajout de ressources. Cette intégration est qualifiée de « légère » ou « lâche » suivant la terminologie introduite par S. Davidson [Davidson, Overton et al. 1995]. La manipulation des données se fait au travers d'une interface de programmation (api) en Java. Dans un premier temps, elle est employée lors de l'intégration des ressources dans l'entrepôt et de son enrichissement. Dans un second temps, des algorithmes permettent l'extraction d'un sous-ensemble contextualisé de l'entrepôt à destination du client. Ces algorithmes sont présents au niveau du serveur, mais sont invoqués par des applications clientes. L'écriture de ces algorithmes ou encore l'interrogation de l'ensemble de l'entrepôt par le client sont possibles au travers de l'api.

Coté client, les besoins de nos interlocuteurs ne se rejoignent pas et ne peuvent être satisfaits efficacement et simultanément dans une seule application. On ne peut envisager une application complexe répondant à la totalité. Nous nous sommes orientés vers la mise à disposition d'une boîte à outils qui permet de construire des clients graphiques riches en fonctionnalités en un minimum de temps, interopérables au travers de la carte. Les éléments clés de la conception sont d'une part la disponibilité de l'interface de manipulation de données, d'autre part la présence de fonctionnalités avancées d'interaction et de visualisation d'information. Cet environnement propose une visualisation (spatiale) permettant à l'utilisateur de se repérer, d'interagir pour explorer, filtrer, annoter les données, réutilisables dans différents contextes : analyse de données d'expression, recherche documentaire, ingénierie des connaissances, etc. Pour évaluer cette approche, nous avons mis en œuvre des expérimentations dans deux

[1] S'agissant d'une introduction, tous ces termes donneront lieu à une définition plus précise dans la suite.

[2] *an Integrated and Interactive Data Exploration Environment*

contextes applicatifs éloignés correspondant aux collaborations que nous avions : l'ingénierie des connaissances et l'analyse de données d'expression. Quelques heures suffisent pour créer un nouveau client bénéficiant de fonctions comme les lentilles sémantiques, zoom et déplacement, panneau d'aperçu, recherche, info-bulles, etc. S'il n'y a pas réellement d'environnement de « script », la spécification de l'interface et de ses fonctionnalités en est proche. Par exemple, il suffit de recopier 5 lignes de code pour ajouter une fonctionnalité de recherche textuelle qui modifie la visibilité et la couleur des éléments répondant au critère de recherche.

La carte des connaissances est donc un outil qui peut se partager au sein de différentes applications, entre différents chercheurs, dans différents domaines scientifiques, etc. La visualisation change fortement d'un contexte à l'autre, et la taille des données reste importante dans de nombreux cas. Il est donc difficile de parler d'une carte mentale réellement partagée par un collectif de chercheurs ou même pour un chercheur seul, dans différents contextes. Cependant, l'utilisation d'une interface commune et somme toute homogène permet d'améliorer le temps d'apprentissage pour l'utilisateur lorsqu'il a déjà une expérience sur une interface graphique cliente. A terme, sa compréhension de la méthode de visualisation qui est assez intuitive peut lui permettre de faire évoluer par lui-même la visualisation

Actuellement, nous développons de nouvelles collaborations avec Yvon Cayre, professeur des hôpitaux de Paris et directeur de l'Unité INSERM 417[1]. Nous avons réalisé avec lui de nouvelles expériences sur puces à ADN dans le contexte de la leucémie promyélocytaire aiguë humaine. Certaines caractéristiques rencontrées lors des expérimentations menées avec Y. Cayre rejoignent nos travaux précédents. Entre autres, il rencontre la difficulté de croiser les données et le besoin d'analyser de grandes quantités de données d'expression.

Plan du mémoire

La **première partie** de ce mémoire introduit quelques notions élémentaires afin de faciliter sa compréhension. Suit alors l'état de l'art correspondant à notre problématique. Le **chapitre 1** introduit la bioinformatique. Il présente les notions élémentaires de biologie moléculaire permettant de comprendre le vocabulaire (ADN, gènes, données d'expression, etc.). Il décrit ensuite la discipline bioinformatique et détaille comment l'informatique est investie dans le quotidien du biologiste. Le **chapitre 2** présente le domaine de l'ingénierie des connaissances et de l'intégration de données. Il définit le vocabulaire du domaine de l'ingénierie des connaissances (terme, concept, relations sémantiques, terminologie, ontologie, etc.). Il met ensuite en évidence la nature hétérogène des données biologiques utilisées par le chercheur et propose un état de l'art du domaine de l'intégration de données. Notre contribution se positionne alors vis-à-vis du domaine de l'intégration de données. Alors que le chapitre 2 aborde l'intégration de données sous l'angle d'une taxonomie technique, le **chapitre 3** approche le problème du point de vue de l'utilisateur final. Quelques expériences montrent comment se traduit l'hétérogénéité pour le chercheur en biologie. L'état de l'art prend du recul sur les aspects sociologiques et techniques de ce domaine dont nous déduisons la nécessité de réconcilier les besoins du développeur et l'utilisateur. Notre approche est ainsi celle de la « *cartographie des connaissances biologiques* ». Nous parcourons les domaines de la cartographie et de la visualisation d'un point de vue théorique et nous positionnons finalement par rapport à l'existant.

La **seconde partie** du mémoire décrit l'environnement que nous avons conçu et mis en œuvre : I^2DEE. Le **chapitre 4** propose une présentation générale de l'architecture d'I^2DEE et de son modèle. Concrètement, un entrepôt de données est implanté coté serveur. Son modèle est en réalité un métamodèle de graphe qui apporte souplesse et extensibilité à I^2DEE et qui lui confère une ouverture aux multiples approches existantes (services Web, systèmes d'intégration, navigateurs, portails, etc.). Coté client, une boîte à outils de visualisation permet de concevoir un

[1] Fonction occupée lors de notre rencontre, cette unité a été dissoute à la fin du plan quadriennal de l'INSERM le 31/12/2006. Y. Cayre est accueilli dans un nouveau laboratoire.

client graphique riche en quelques heures, adapté à un contexte applicatif spécifique. Le **chapitre 5** présente la construction de l'entrepôt, et plus précisément les procédures d'intégration de données, de fouille de textes et d'indexation. Le **chapitre 6** présente la boîte à outils graphique d'I^{2}DEE basée sur Prefuse et que nous avons fait évoluer de façon importante en poursuivant l'objectif de fournir à l'utilisateur des fonctions de haut niveau rapides à implémenter pour un développeur peu expérimenté dans la conception d'interfaces utilisateur. La fin de ce chapitre est dédiée aux mécanismes d'adaptabilité permettant d'extraire une carte contextuelle de l'entrepôt et de l'adapter au besoin de l'utilisateur en apprenant de ses interactions notamment. Le **chapitre 7** présente les résultats visuels obtenus dans deux contextes applicatifs : la conception de ressources terminologiques et ontologiques et l'analyse de données d'expression de gènes issues de puces à ADN. Enfin, le **chapitre 8** conclut sur notre contribution, sa réponse à la problématique initiale et discute des grandes directions de recherche qui nous motivent : dans un premier temps, nous souhaitons mettre en œuvre l'environnement dans une expérience complète à dimension réelle et évaluer les bénéfices apportés par I^{2}DEE. Dans un second temps, nous pensons à apporter plus de souplesse et de polyvalence à l'environnement en confiant plus de responsabilités aux scripts dans un contexte plus ouvert.

Partie 1

Prérequis, Problématique & Etat de l'art

CHAPITRE 1

Mise en contexte de notre problématique

« Ma carrière a suivi une descente des dimensions les plus grandes aux dimensions les plus petites, avec le désir de comprendre la vie. Je suis ainsi passé des animaux aux cellules, des cellules aux bactéries, des bactéries aux molécules, des molécules aux électrons. Mais l'ironie de cette histoire est que les molécules et les électrons n'ont précisément pas de vie. Dans ma démarche, la vie m'a comme filé entre les doigts. Il me faut désormais retourner sur mes pas, en remontant cet escalier que j'ai eu tant de peine à descendre. »

ALBERT SZENT-GYÖRGYI
LAUREAT DU PRIX NOBEL DE MEDECINE EN 1937

1.1 Introduction

On regroupe sous le terme de bioinformatique un champ de recherche multidisciplinaire où travaillent de concert biologistes, informaticiens, mathématiciens et physiciens, dans le but de résoudre un problème scientifique posé par la biologie. » (Wikipédia).

Il s'agit de rassembler des communautés des sciences expérimentales (biologie, biochimie, etc.) et des sciences formelles (informatique, mathématique et statistiques, etc.) dans le but de favoriser des avancées simultanées dans les différents domaines. Un point de vue plus restrictif est adopté dans la définition de Jean-Michel Claverie[1] :

La Bioinformatique est la discipline de l'analyse de l'information biologique, en majorité sous la forme de séquences génétiques et de structures de protéines. C'est une branche théorique de la Biologie, largement antérieure à la récente "révolution génomique". Malgré son nom, la "bioinformatique" ne doit pas être confondue avec une simple application aux données biologiques des concepts et des outils de l'informatique traditionnelle.

Cette définition est l'illustration d'un constat fréquent : les travaux contribuent individuellement de façon déséquilibrée aux deux disciplines : les informaticiens, statisticiens et mathématiciens proposent de nouvelles méthodes ou de nouveaux outils. Les biologistes en tant qu'utilisateur contribuent à de nouvelles connaissances dans leur discipline propre et permettent d'améliorer les méthodes par leurs retours d'expériences. L'informatique représente pour eux un dispositif expérimental *in silico,* au même titre que pour d'autres un dispositif *in vivo* ou *in vitro.*

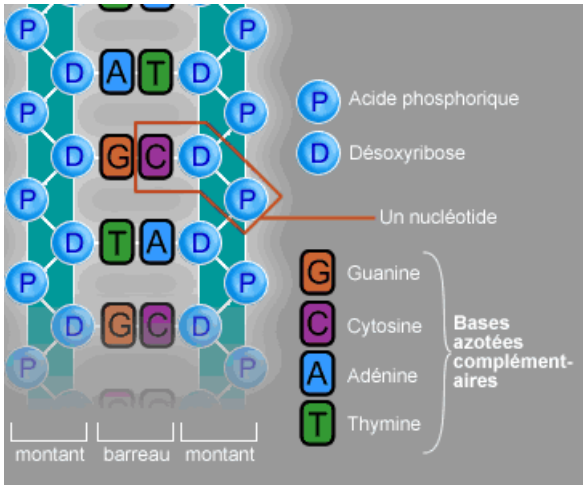
L'informatique est plus généralement présente dans le domaine biomédical. Pour certains, la bioinformatique n'est pas incluse dans l'informatique biomédicale. Pour d'autre, c'est le cas. EN témoignent des ressources et des institutions communes comme PubMed, Entrez, ou UMLS qui dépendent du NIH. Le rapport établi par le consortium européen Infobiomed défini la bioinformatique et la médicoinformatique comme deux disciplines fondamentales de l'informatique biomédicale [Potamias 2006].

Cette thèse est celle d'un informaticien désireux d'améliorer le quotidien de chercheurs en biologie moléculaire et cellulaire en répondant aux besoins d'intégration et de visualisation de l'information biologique. Nous utilisons des ressources provenant du domaine biomédical dans son ensemble et nous nous inspirons des contributions apportées par ces différentes communautés. Il est tout à fait envisageable de transposer ce travail dans un contexte biomédical, et même plus largement encore, du moment que les problématiques restent similaires : intégrer et visualiser des données hétérogènes et volumineuses. Si l'accent est mis, dans ce mémoire, sur la biologie, c'est le biomédical dans sa globalité qui a contribué à notre travail. Dans la suite, nous employons alternativement les deux termes, biologie et médical, cependant cette distinction communautaire n'est pas synonyme d'une limite de l'adaptabilité dans l'une ou l'autre des deux communautés.

Ce chapitre présente quelques notions élémentaires de biologie pour l'informaticien qui n'est pas familier de ce domaine. Après un bref rappel de biologie moléculaire (ADN-ARN-protéine). La technologie des puces à ADN est présentée à partir d'un scénario. Nous verrons en particulier l'intérêt pour le biologiste de disposer d'interfaces visuelles. Nous décrivons enfin au travers de l'analyse de données d'expression l'implication des outils informatiques dans le quotidien du chercheur en sciences de la vie. En particulier, nous présentons les systèmes d'information qu'il consulte au travers de quelques exemples liés à notre contexte (*Plasmodium Falciparum* et la leucémie promyélocytaire aiguë).

[1] La bioinformatique : une discipline stratégique – http://www.igs.cnrs-mrs.fr/SpipInternet/article.php3?id_article=178

1.2 Notions élémentaires de biologie moléculaire



▲ *Figure 1.1 – Composition d'un brin d'ADN [W]*[1]

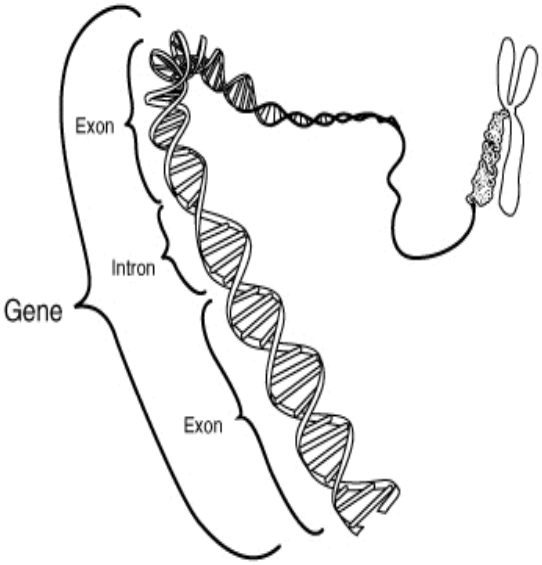


Figure 1.2 – Structure en double hélice de l'ADN [W] ►

1.2.1 Généralités : ADN, ARN et protéines

L'ADN (acide désoxyribonucléique) est le support de l'hérédité dans nos cellules. Il s'agit d'une longue séquence de nucléotides dans laquelle on distingue 4 nucléotides qui diffèrent par leur base azotée (figure 1.1). Un brin d'ADN peut être considéré comme un long mot construit suivant un alphabet de 4 lettres : A (adénine), T (thymine), C (cytosine) et G (guanine). Il peut atteindre des longueurs de l'ordre de plusieurs millions de nucléotides. A/T et C/G forment deux paires de bases nucléotidiques complémentaires suivant deux ou trois liaisons hydrogène. L'ADN est généralement présent sous forme de double brin et adopte une structure de double hélice, qui chez certains organismes se compacte pour former un chromosome (figure 1.2). Cet ADN est répliqué de façon identique dans toutes les cellules d'un même être vivant suivant le principe schématisé sur la figure 1.3.

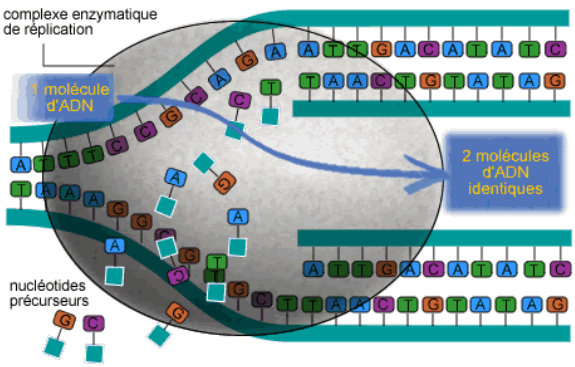


Figure 1.3 – Réplication de l'ADN [W]

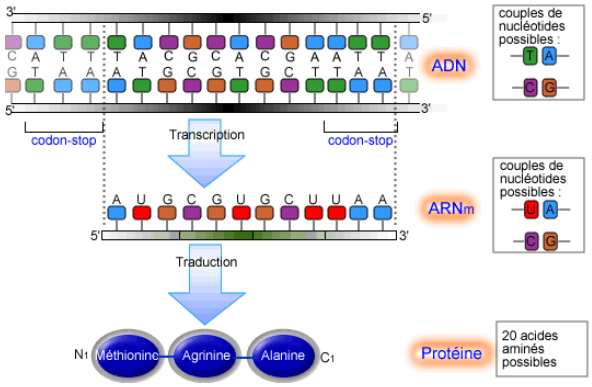


Figure 1.4 – ADN, ARN et Protéine [W]

[1] *Dans cette section, les illustrations ont été empruntées à plusieurs ressources. Une grande partie des schémas généraux provient de l'encyclopédie en ligne libre* Wikipédia *et est soumise à une licence* dfdl. *Les photographies et illustrations concernant les puces à* ADN *sont gracieusement mises à disposition par la société* Affymetrix. *Nous indiquons respectivement par un* [W] *ou* [Aff.] *les figures provenant de ces deux sources.*

L'ADN n'est pas directement impliqué dans l'activité de la cellule. Il code les protéines qui interviennent dans le fonctionnement de la machinerie cellulaire. Une protéine peut être définie comme une chaîne composée à partir de 20 acides aminés. Le passage de l'ADN aux protéines n'est pas direct (figure 1.4), l'ADN est d'abord transcrit en ARN messager (ARNm – figure 1.5), acide ribonucléique. Ce dernier est similaire à l'ADN : le sucre désoxyribose est remplacé par un ribose, et la thymine est remplacée par l'uracile (toujours complémentaire de l'adénine). L'ADN qui stocke l'information génétique est séquestré dans le noyau. L'ARNm, réplique de l'ADN, peut quitter le noyau de la cellule afin d'être traduit en protéine. Contrairement à l'ADN nucléaire, l'ARN est fragile, se dégrade rapidement, et constitue des molécules plus courtes.

L'ARNm est par la suite traduit en protéine au travers d'un mécanisme associant à un codon d'ARN (triplet de nucléotides) un acide aminé (figure 1.6). Cette correspondance est appelée le code génétique (figure 1.8). Certains codons sont appelés *« start »* et *« stop »* car ils ouvrent ou ferme un cadre de lecture (« ORF – *Open reading frame* »), c'est-à-dire une région codante qui donnera lieu à une transcription. Un gène est une séquence génétique fonctionnelle. Elle comporte une partie codante (et donc un codon *start* et un codon *stop*), mais aussi une partie amont non codante qui régule l'expression du gène[1].

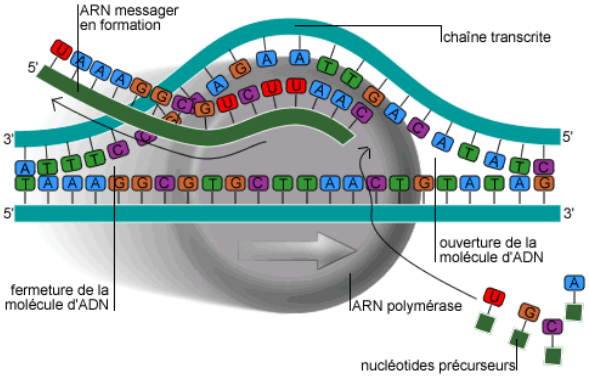


Figure 1.5 – Transcription de l'ADN en ARN [W]

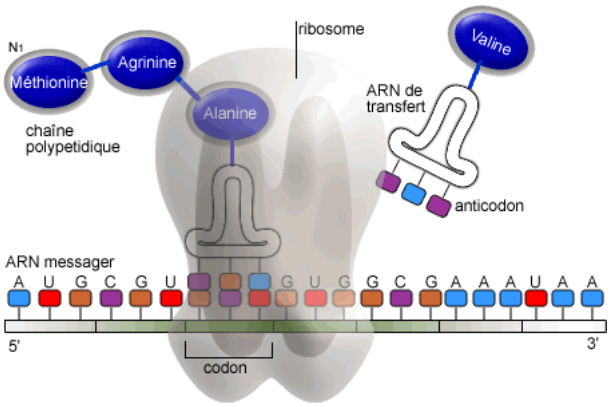


Figure 1.6 – Traduction de l'ARNm en protéine. [W]

La transition de l'ADN à la protéine n'est pas aussi simple que nous l'avons illustrée auparavant. En réalité, différents mécanismes intermédiaires sont mis en jeu :

- l'ADN est transcrit en pré-ARNm (figure 1.5),
- chez les eucaryotes, le pré-ARN est alors *épissé* (figure 1.7): durant cette étape, certaines parties d'un gène sont excisées (épissage alternatif). Ce mécanisme complexe permet à partir d'un même gène de produire plusieurs protéines différentes (mais pas simultanément), dites homologues. A l'issue de ce processus, on obtient de l'ARNm (dit mature ou messager).
- chez les procaryotes, certains gènes sont dits *polycistroniques* ; ils codent plusieurs protéines qui participent à une même fonction.
- enfin, l'ARNm est directement traduit en séquence d'acides aminés (figure 1.7) à partir du code génétique (figure 1.8).

Une protéine se replie en trois dimensions de façon unique pour un milieu défini. Sa configuration tridimensionnelle détermine ses interactions avec d'autres complexes biochimiques. Cette configuration est directement responsable de sa fonction biologique.

Afin de caractériser le dispositif expérimental d'un chercheur, son positionnement dans ces différentes étapes de la biologie moléculaire, nous utilisons certains termes généraux : génomique, transcriptomique, protéomique. La section suivante explique ces termes et y présente l'apport de l'informatique.

[1] Cette définition est simplifiée et suffisante pour la compréhension de ce mémoire.

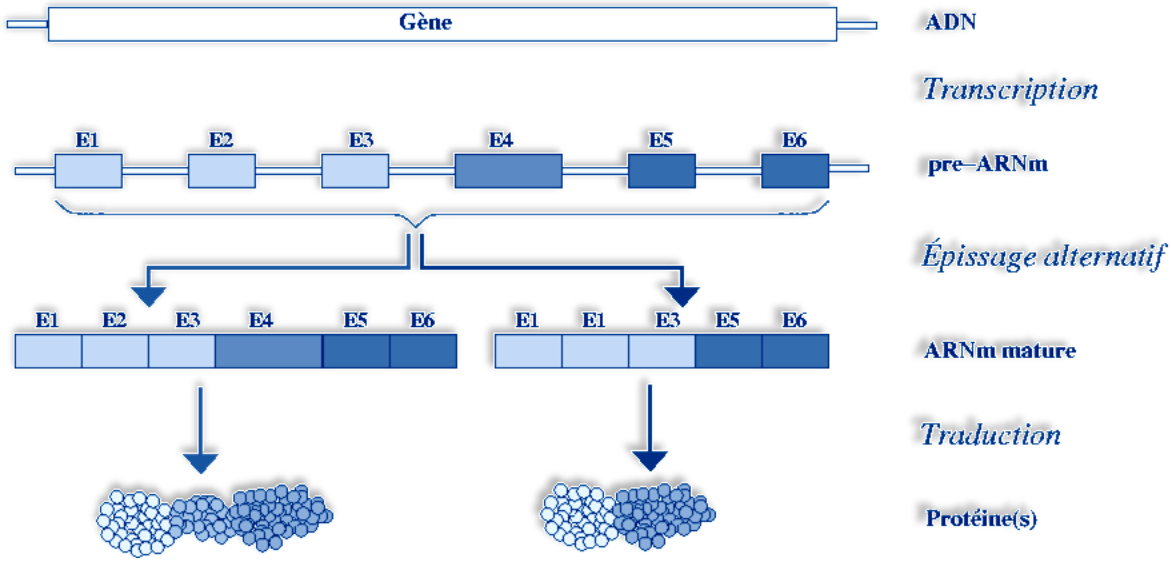


Figure 1.7 – Transcription et épissage alternatif [W]

	U			A			G			C		
U	UUU	*Phe/F*	*Phenylalanine*	UCU	*Ser/S*	*Serine*	UAU	*Tyr/Y*	*Tyrosine*	UGU	*Cys/C*	*Cysteine*
	UUC			UCC			UAC			UGC		
	UUA	*Leu/L*	*Leucine*	UCA			UAA	***Ochre***	***Stop***	UGA	***Opal***	***Stop***
	UUG			UCG			UAG	***Amber***		UGG	*Trp/W*	*Tryptophan*
C	CUU	*Leu/L*	*Leucine*	CCU	*Pro/P*	*Proline*	CAU	*His/H*	*Histidine*	CGU	*Arg/R*	*Arginine*
	CUC			CCC			CAC			CGC		
	CUA			CCA			CAA	*Gln/Q*	*Glutamine*	CGA		
	CUG			CCG			CAG			CGG		
A	AUU	*Ile/I*	*Isoleucine*	ACU	*Thr/T*	*Threonine*	AAU	*Asn/N*	*Asparagine*	AGU	*Ser/S*	*Serine*
	AUC			ACC			AAC			AGC		
	AUA			ACA			AAA	*Lys/K*	*Lysine*	AGA	*Arg/R*	*Arginine*
	AUG	*Met/M*	*Methionine*	ACG			AAG			AGG		
G	GUU	*Val/V*	*Valine*	GCU	*Ala/A*	*Alanine*	GAU	*Asp/D*	*Aspartic acid*	GGU	*Gly/G*	*Glycine*
	GUC			GCC			GAC			GGC		
	GUA			GCA			GAA	*Glu/E*	*Glutamic acid*	GGA		
	GUG			GCG			GAG			GGG		

Figure 1.8 – Le code génétique : correspondance entre un codon d'ARN un acide aminé.

1.2.2 Définitions : génomique, transcriptomique, protéomique, …

Lorsque l'on décrit des travaux en biologie, on utilise fréquemment trois termes généraux permettant de les positionner : la génomique, la transcriptomique et la protéomique. La **génomique**, est l'étude du génome, c'est-à-dire l'ensemble des gènes d'un ou plusieurs organismes. Une hypothèse de la génomique est basée sur la similarité de séquences : deux séquences similaires partagent des propriétés fonctionnelles.

Les outils bioinformatiques mettent à disposition des algorithmes alignant des séquences, et proposant différentes distances, mesures de similarités, etc. La génomique utilise alors ces outils avec deux principales finalités :

- la phylogénie consiste à construire un arbre de l'évolution des organismes vivants à partir des similarités et des différences entre leurs génomes [Guindon 2003].

- la génomique fonctionnelle, au contraire de la phylogénie, ne s'intéresse pas au génome globalement, mais plus particulièrement au gène. Elle vise à améliorer la compréhension des gènes : leurs fonctions, leur régulation, leurs interactions, etc. Par exemple, la fonction d'un gène permet de prédire la fonction d'un gène similaire dans une autre espèce en se basant sur le principe de l'évolution.

La **transcriptomique** s'intéresse aux gènes transcrits, c'est-à-dire aux ARN. L'ADN est répliqué de façon identique dans toutes les cellules de notre organisme. La présence de l'ARN au contraire évolue avec le temps, suivant la fonction de la cellule, etc. Il est traduit par la suite en protéines dans la cellule. La transcriptomique mesure l'expression des gènes, c'est-à-dire la quantité d'ARNm présente dans une cellule à un instant donné. Elle étudie aussi les mécanismes de transcription et d'épissage.

Enfin, la **protéomique** s'intéresse aux protéomes, c'est-à-dire à l'ensemble des protéines traduites qui sont présentes dans une cellule. La diversité des protéines est nettement supérieure à celles des ARN.

Ce cloisonnement en trois sous spécialités d'une communauté n'est bien sur pas suffisant pour représenter l'activité réelle d'un chercheur. Un biologiste qui étudie un gène peut avoir pour objectif final d'en connaître la fonction, c'est-à-dire savoir quelle protéine est traduite, avec quelles molécules elle interagit, etc. Il existe de nombreux autres champs d'études: le métabolome qui s'intéresse à toutes les petites molécules, les voies de signalisation, les voies métaboliques, la physiologie, etc.

1.2.3 Données d'expression et analyse haut-débit.

Le patrimoine génétique d'un être vivant est identique dans chacune de ses cellules. Pour autant, toutes les cellules n'ont pas la même fonction, et les protéines n'y sont pas présentes uniformément. Les données d'expression des gènes quantifient la présence de l'ARNm et par conséquent l'activité de synthèse des protéines. Il existe différentes technologies haut-débit permettant de mesurer l'expression de gènes. Parmi elles, les puces à ADN sont souvent employées, notamment par les biologistes avec lesquels nous collaborons. Il existe plusieurs procédés de fabrications de ces puces et les progrès technologiques ont permis en quelques années une forte augmentation des densités et par conséquent du nombre de gènes analysés simultanément sur une puce. Toutes ces technologies et leur variantes reposent cependant sur un principe maintenant ancien de *« Southern blot »* [Southern 1975]. Dans la suite, le procédé de fabrication et d'analyse d'une puce à ADN sur lame de verre est présenté [DeRisi, Vishwanath et al. 1997] (cf. figure 1.10). Quelques précisions sur les différentes technologies sont reprises à la fin de cette section, et les spécificités relatives aux puces que nous manipulons sont présentées dans la section suivante. Pour plus d'information sur les différents procédés de conception et de puce et sur leur analyse, nous recommandons la lecture de la thèse de Waka Lin [Lin 2004].

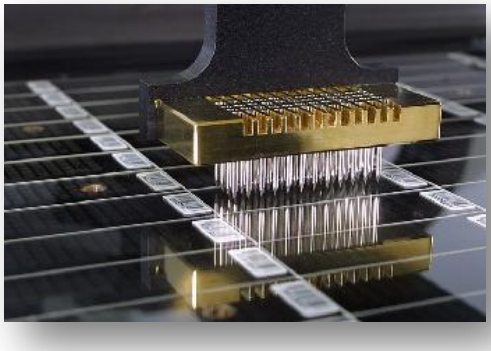


Figure 1.9 – Image d'un « spotter » : le robot dépose à l'aide d'aiguilles très fines les sondes sur les lames de verre avec une grande précision. [1]

Figure 1.10 – Une puce à ADN sur lame de verre. Elle possède une telle densité qu'on ne voit rien à l'œil nu. [2]

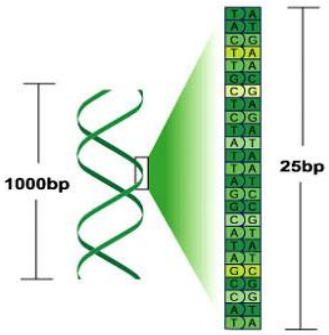


Figure 1.11 – Une sonde est une courte séquence identifiant un gène [Aff.].

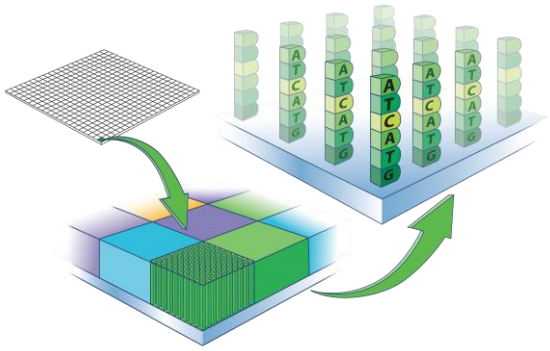


Figure 1.12 – Chaque position de la matrice regroupe des sondes d'ADN simple brin. [Aff.]

L'objectif d'une puce est d'analyser l'expression de gènes dans des cellules. Une sonde nucléique est un oligonucléotide[3] qui identifie un gène donné (figure 1.11). Pour concevoir la puce, on commence donc par déposer des sondes sur un support[4] (cf. figure 1.9). Chaque « case » du support regroupe des sondes identiques, identifiant le plus souvent un gène, un ORF, etc. (figure 1.12). En parallèle, on prépare l'échantillon de matériel biologique que l'on souhaite étudier. On extrait les ARNm des cellules que l'on rétrotranscrit en ADNc[5]. Les brins d'ADN retrotranscrits sont simultanément marqués par un fluorochrome[6].

Une fois ce matériel biologique prêt, on le dépose sur la puce. Lorsque l'on ajoute l'échantillon sur la puce, les brins d'ADN s'hybrident (se fixent) avec les sondes complémentaires (figure 1.13). La puce est finalement rincée ; Le matériel biologique qui ne s'est pas fixé est éliminé. Les brins qui restent (hybridés) comportent un marqueur fluorescent (figure 1.14) et correspondent donc aux sondes initialement déposées sur membrane. Les gènes absents n'ont pas été hybridés, ils ne seront par conséquent pas fluorescents sur la puce. Un spot est donc d'autant plus lumineux que les sondes lui correspondant se seront hybridées avec l'échantillon

[1] Photo: National Research Council of Canada

[2] Photo : Agilent Technologies

[3] Courte séquence nucléotidique dont la longueur est de l'ordre de 12 à 70 bases en général

[4] Ici il s'agit d'une lame de verre. Ailleurs, on utilise des membranes de nylon par exemple.

[5] Procédure inverse de la transcription de synthèse d'un brin d'ADN complémentaire d'un brin d'ARN.

[6] Il existe d'autres types de marquage. Par exemple, le *Southern Blot* utilise un marquage radioactif.

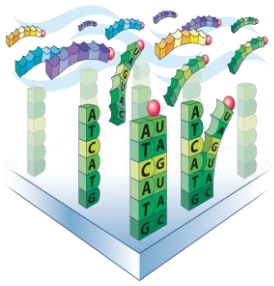


Figure 1.13 – L'ADN rétrotranscrit est hybridé[1]. [Aff.]

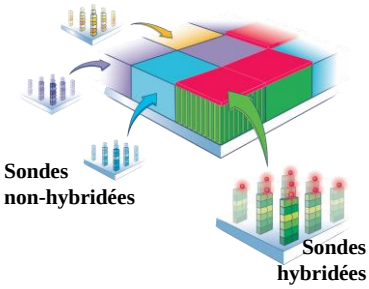


Figure 1.14 – Après le rinçage, seule les sondes reconnues sont fluorescentes. [Aff.]

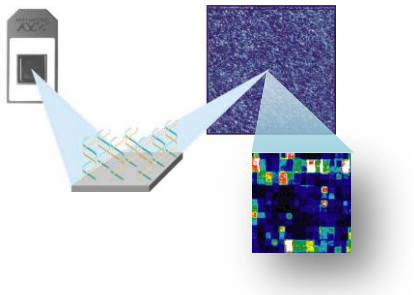


Figure 1.15 – Exemple d'image produite par une puce à ADN [AFF.]

La puce est alors analysée par un scanner qui produit une image et indique la valeur de fluorescence pour chaque position de gène[2] (cf. figure 1.15). De multiples problèmes peuvent subvenir et produire une image difficile à analyser, si l'aiguille est trop éloignée du support, ou au contraire rentre en contact avec lui (éclaboussement du voisinage, aiguille abîmée pour les futurs dépôts, etc.). L'homme intervient pour corriger et assister le détourage des *spots* proposé par un logiciel d'analyse d'image. Des prétraitements statistiques sont appliqués afin de transformer le signal lumineux en valeur significative, détecter et corriger les différents aléas expérimentaux.

Les puces ADN n'ont de rapport avec les circuits intégrés que concernant la notion de miniaturisation. En effet, une puce n'est utilisable qu'une seule fois et elle s'avère très coûteuse[3]. Il existe différents procédés de fabrication et de fonctionnement des puces, qui ont des impacts importants mais respectent le principe général présenté précédemment. Affymetrix, synthétise les sondes sur un support carré de 12,8 mm de coté par un procédé appelé photolithographie reposant sur l'utilisation d'un masque. Cette technologie permet d'obtenir une densité importante de 250 000 spots par puce[4] (soit un spot de près de 20µm). Leur technologie repose aussi sur l'emploi d'un seul marqueur de fluorescence. Les puces sur lame de verre les plus courantes sont construites par le dépôt robotisé des sondes sur le support. La taille d'un spot n'est alors que de 100 µm ce qui permet d'analyser simultanément quelques dizaines de milliers de gènes sur une lame. En général, les puces lames de verre utilisent simultanément deux fluorochromes de couleur différente, le Cy3 (vert) et le Cy5 (rouge). Deux échantillons biologiques différents sont préparés et analysés simultanément. Ceci permet de réaliser une analyse différentielle plus simplement ou d'inclure un contrôle de qualité. Le scanner utilise alors deux longueurs d'onde différentes. Ce principe général de fonctionnement d'une puce à ADN est résumé dans la figure 1.16 (à gauche) et un exemple d'image produite par le scanner est illustré dans la même figure, à droite.

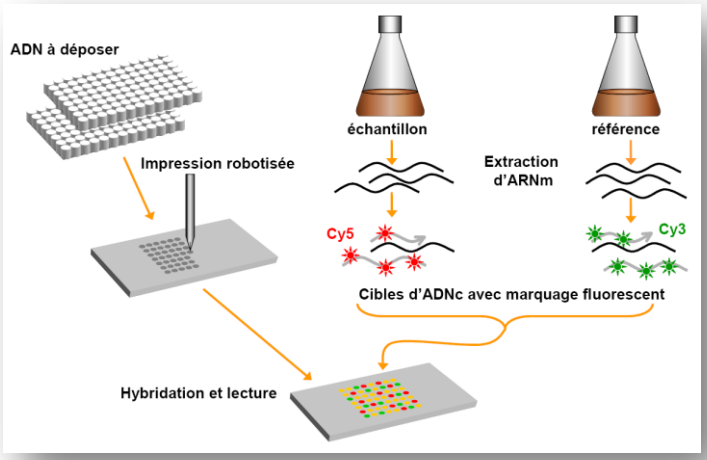
D'autres technologies permettent d'obtenir des données d'expression. Ainsi des recherches sont en cours pour créer des puces à protéines (mesurant l'expression des protéines). Mais d'autres techniques sont utilisées depuis plusieurs années. L'équipe de Jacques Bourguignon (CEA) dans le contexte du projet *Arabidopsis* mesure la quantité de protéines présente à l'aide d'une spectrométrie de masse haut-débit [Sarry, Kuhn et al. 2006].

[1] Sur ce schéma d'Affymetrix, l'hybridation a lieu entre ADN et ARNm, il s'agit d'une particularité de leur technologie.

[2] On parle alors de *spot* de gène.

[3] Le coût d'une puce peut dépasser souvent le millier d'euros, les investissements sont aussi importants.

[4] A titre de comparaison, on estime que le génome humain comporte autour de 30 000 gènes. Cela permet à Affymetrix d'utiliser une combinaison de 4 spots pour augmenter la fiabilité de l'analyse d'un gène.



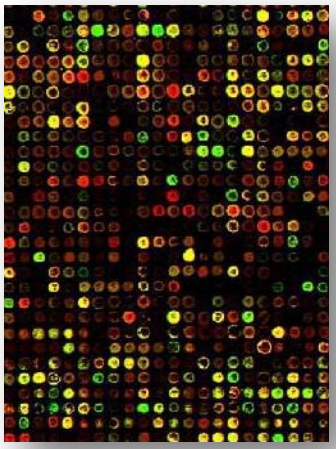
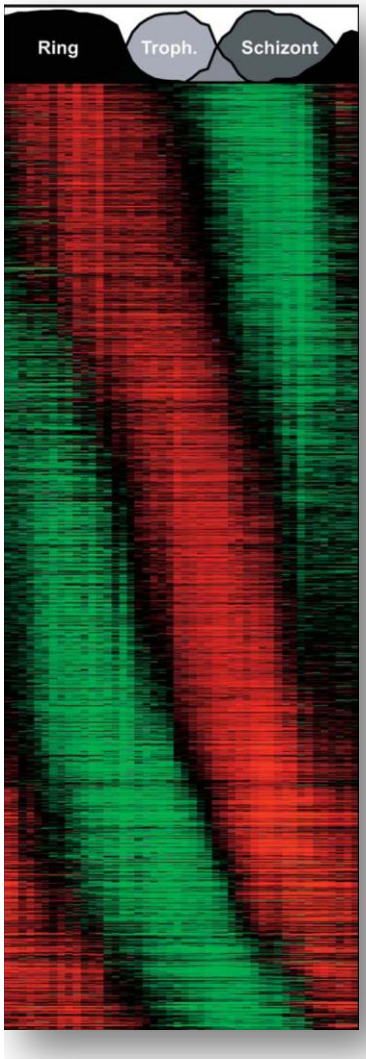


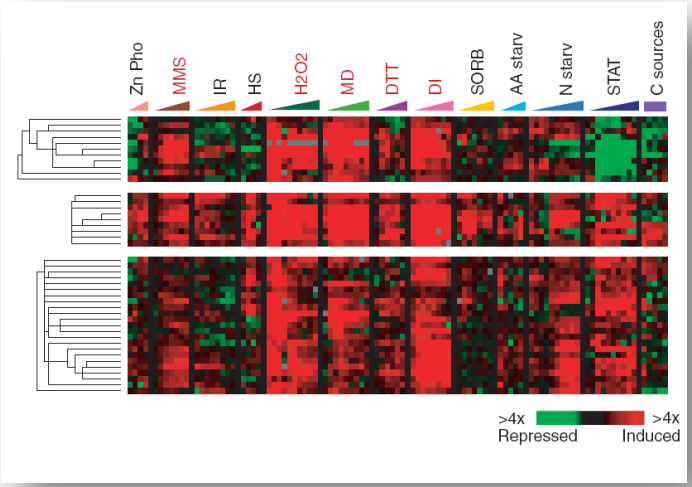
Figure 1.16 — Le schéma à gauche [Lin 2004] résume le principe d'utilisation de deux marqueurs fluorescents de couleurs différentes sur une lame de verre. On dépose sur un support les sondes. En parallèle, on extrait les ARNm des deux échantillons en culture que l'on rétrotranscrit en ADNc. Chaque échantillon est marqué par un fluorochrome différent (Cy3 pour le vert, Cy5 pour le rouge). Les échantillons sont placés sur la puce (hybridation). La puce est rincée puis analysée à l'aide d'un scanner. L'image produite est comparable à celle présentée à droite [W]. Un spot vert signifie la présence de l'échantillon marqué au Cy3, un spot rouge de celui marqué au Cy5, un spot jaune la présence des deux, et un spot noir l'absence des deux.

Les puces peuvent être employées dans différentes perspectives comme le diagnostic ou le pronostic. Dans ces cas, une puce unique est employée pour chaque patient. D'autres expériences sont différentielles : on étudie la variation de l'expression en variant les conditions expérimentales. Il faut alors une puce par condition. Certaines analyses sont temporelles, une puce est utilisée à chaque instant où l'on souhaite mesurer l'expression des gènes. Ces analyses sont généralement coûteuses et nécessitent un grand nombre de puces. Dans le contexte de l'étude de *Plasmodium Falciparum*, une expérience qui fait référence est celle menée par Zbynek Bozdech [Bozdech, Llinás et al. 2003] qui mesure l'expression des gènes toutes les heures pendant le cycle intraérythrocytaire (48 heures – figure 1.17). Pour que ce type d'expérience réussisse, il faut que toutes les cellules d'un échantillon déposé sur une puce soient synchronisées au même stade de la vie cellulaire et d'un cycle. Il existe enfin des approches mixtes : [Gasch and Eisen 2002] étudie par exemple l'évolution temporelle de la levure suivant 13 conditions différentes (figure 1.18). Dans tous ces cas, il est indispensable de réaliser des réplicas, c'est-à-dire des mesures supplémentaires permettant d'augmenter la fiabilité. Ceci alourdit le coût d'une expérience.



◀ *Figure 1.17 – Résultat visuel de données d'expression issu de [Bozdech, Llinás et al. 2003]. Cette matrice de points bicolore est couramment utilisée pour visualiser des données d'expression. Une ligne est un vecteur (gène), une colonne une mesure (puce). Les couleurs n'ont pas la même signification que celle d'une image directement issue d'un scanner résultant de l'emploi de deux colorants. Un point vert signifie la surexpression d'un gène, un point rouge sa répression. Ce type de visualisation, bien que souvent employé est généralement décrit comme peu compréhensible par leur auteur, mais dans certains cas particuliers, elles font apparaître des motifs. Ici par exemple, elle met en évidence la cyclicité de la vie de* Plasmodium Falciparum.

▼ *Figure 1.18 – Diagramme d'Eisen issu de [Gasch and Eisen 2002]. Ce type de diagramme combine une matrice d'expression similaire à la précédente et un dendrogramme résultant d'une classification ascendante hiérarchique des gènes. Les colonnes peuvent aussi parfois être réordonnées et associées à un dendrogramme. Ici, chaque triangle situé au dessus de la matrice représente une condition expérimentale différente évoluant dans le temps.*



Les données d'expression sont des données vectorielles et sont généralement représentées sous forme de matrice de points verts et rouges. Cette représentation reprend les couleurs des deux marqueurs les plus courants en biologie, mais leur signification n'a en réalité aucun rapport : le vert indique un gène réprimé, le rouge un gène surexprimé, le noir une valeur intermédiaire. L'échelonnage des couleurs ne va pas au-delà de cette codification et change d'une publication à l'autre (en général il est indiqué en légende). Chaque ligne est un gène (ou tout au moins un spot sur la puce, à un gène peuvent correspondre plusieurs spots), chaque colonne une mesure (une puce[1]). En général, après l'obtention de ce résultat on applique une classification hiérarchique sur les gènes, produisant un dendrogramme[2] (dessiné sur la gauche dans la figure 1.18). Lorsque les conditions expérimentales ne sont pas liées aux temps ou à un ordre d'intensité dans les conditions expérimentales, il arrive fréquemment que l'on réordonne les colonnes à l'aide d'une classification hiérarchique et que l'on dessine aussi un dendrogramme pour ces colonnes.

[1] Dans le cas où l'usage de deux colorants donnerait lieu à deux mesures, on peut envisager que deux colonnes soient associées à une seule puce.

[2] Un dendrogramme est un arbre enraciné dont la longueur entre la racine et toutes les feuilles est identique. La longueur qui sépare deux nœuds correspond à une mesure de similarité ou de distance.

Les différentes techniques que nous venons de présenter produisent de très nombreuses données qu'il est nécessaire d'analyser, de croiser, d'interpréter. Les outils informatiques permettent d'automatiser une partie de ces tâches ou tout au moins assistent le biologiste qui en a la charge.

1.3 Biologie *In silico*

Ces nouvelles technologies haut-débit ont installé l'informatique dans le quotidien du biologiste. L'omniprésence de l'informatique est telle qu'on parle aujourd'hui de *biologie virtuelle*, ou encore d'un dispositif expérimental *in silico*, au même titre que les dispositifs *in vivo* et *in vitro*. Ainsi, certaines contributions sont issues essentiellement de manipulations informatiques : c'est le cas de nombreuses études menées en phylogénétique [Guindon 2003], des travaux de fouille de texte découlant de l'initiative de Don R. Swanson [Swanson 1986], ou encore expérimentations basées sur un environnement de simulation cellulaire [Tomita, Hashimoto et al. 1999; Neyfakh, Baranova et al. 2006]. Alors que certains problèmes ne motivent qu'un nombre restreint de biologistes, d'autres tels que la recherche d'information concernent la majorité d'entre eux, dans la plupart de leurs tâches : état de l'art, publication, analyse de données et résultats expérimentaux, mise au point de protocoles expérimentaux, etc.

Dans la suite de cette section nous illustrons notre propos en détaillant une expérience dans laquelle nous avons fortement été impliqués. Nous décrivons globalement la démarche expérimentale en signalant systématiquement l'usage de l'informatique dans chacune des étapes. Nous n'avons pas été impliqués dès le début des expérimentations menées par l'institut Pasteur, et certains points sont confidentiels. Nous prenons comme exemple une expérience plus récente résultat de notre collaboration avec Yvon Cayre, concernant la leucémie promyélocytaire aiguë (chez l'homme). Les expériences ont été menées sur une plateforme de puces à ADN fabriquée par Applied Biosystems (figure 1.19). Ce choix provient de l'équipement préexistant dans le laboratoire de nos collaborateurs.

Figure 1.19 – Photographie d'une plateforme de puces à ADN d'Applied Biosystems (AB). A gauche une puce, ses dimensions sont plus grandes qu'une lame de verre : Le diamètre de la membrane (au centre) est voisin de 7 cm. A droite, la photo montre le scanner.

1.3.1 Un scénario d'expérience sur des puces à ADN

Le projet dans lequel nous sommes impliqués concerne la leucémie promyélocytaire aiguë. Plus particulièrement, son objectif est d'étudier l'effet de l'acide rétinoïque sur des cellules présentant une mutation caractéristique. L'acide rétinoïque a des effets thérapeutiques temporaires sur cette pathologie, le mode d'action étant identifié au niveau cellulaire. Des études existent déjà dans des contextes similaires, cependant, nous souhaitons reprendre les expérimentations en totalité afin d'ôter tout *a priori.* Ces expériences ont pour objectif de mettre en évidence des gènes d'intérêt. Pour une analyse globale du génome, les puces à ADN sont donc

privilégiées. La démarche de l'expérience est de mesurer l'expression des gènes en présence ou non d'acide rétinoïque et de comparer les données d'expression. On recherche alors les gènes ayant une grande différence d'expression (analyse comparative ou différentielle).

Pour en arriver à cette problématique, il a été nécessaire d'effectuer un état de l'art et de se renseigner sur les travaux voisins. Ce travail est antérieur à notre collaboration, et résulte de l'expérience acquise par les chercheurs et le laboratoire pendant des années. Un autre héritage de ce laboratoire est la plateforme de puces à ADN d'Applied Biosystems (figure 1.19). Dans le cas où un système intégré comme celui-là n'existe pas, le biologiste peut être amené à produire une liste de gènes à étudier, rechercher leurs séquences et déterminer à partir de cela les séquences des sondes qui seront nécessaires et les commander. Dans chacune de ces tâches, l'informatique est indispensable. Dans notre cas, la quantité d'échantillons et de réactifs était réduite, mais il fallait tout de même organiser les données au travers d'un logiciel. Toute la planification de l'expérience est décrite dans un logiciel spécialisé favorisant une bonne traçabilité de ces expériences, permettant de spécifier les différents paramètres (températures d'hybridations, concentration des réactifs, durées de centrifugation, d'incubation, etc.).

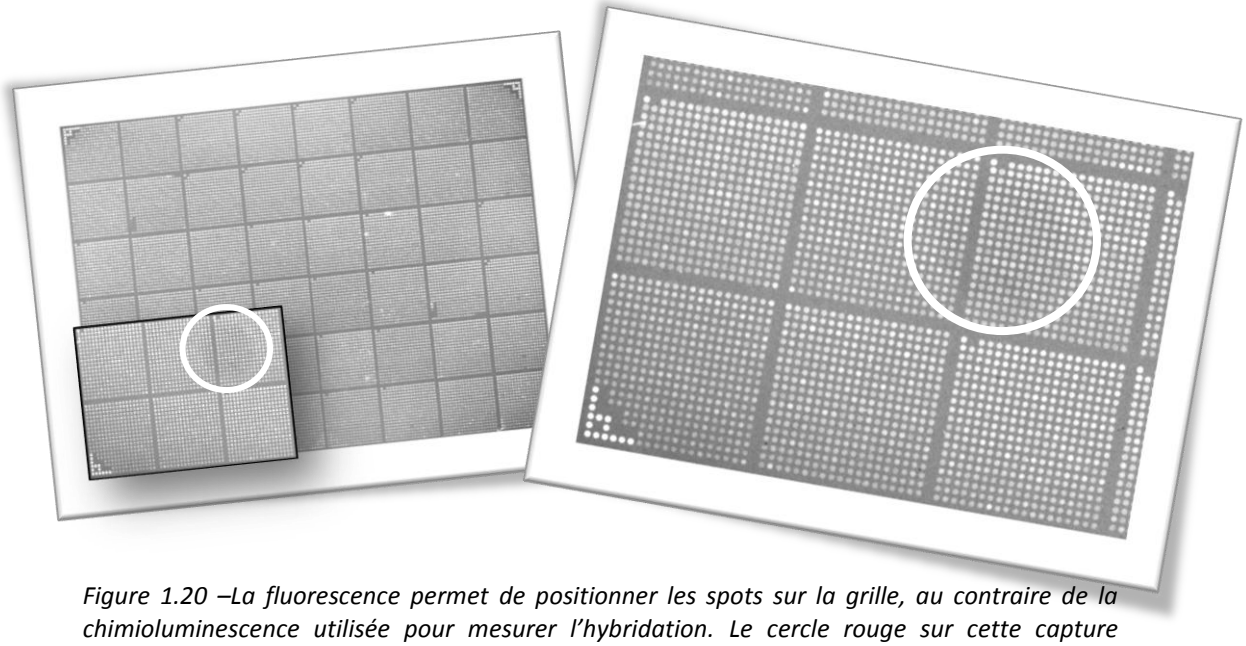


Figure 1.20 –La fluorescence permet de positionner les spots sur la grille, au contraire de la chimioluminescence utilisée pour mesurer l'hybridation. Le cercle rouge sur cette capture montre qu'il y a une auréole foncée résultant d'un aléa. La fluorescence est une information complémentaire pour la fiabilité des données.

Une fois l'expérience entièrement planifiée, sa mise en œuvre ne nécessite pas systématiquement la présence de l'informatique. Cela dépend de l'appareillage utilisé, mais l'informatique est généralement présente dans les dispositifs haut-débits. Dans le contexte de nos expérimentations, ce n'est qu'à la fin du dispositif que nous y avons recours, du fait de la présence d'un scanner. Pour l'analyse d'une puce, 8 images au format TIFF sont produites. Le scanner est équipé d'une caméra CCD haute définition. Chaque puce est photographiée plusieurs fois. La puce est découpée en 2 régions, et l'appareil photo prend un cliché pour chacune des deux moitiés. De plus, afin de bien traiter les luminescences fortes et faibles, deux vitesses d'obturation sont utilisées (5 sec. d'exposition pour les *spots* lumineux, 25 sec. pour les signaux plus faibles). Enfin, on utilise un filtre optique pour séparer la fluorescence de la chimioluminescence. La fluorescence est employée pour le positionnement des spots et le contrôle de la qualité (figure 1.20). La chimioluminescence est utilisée pour mesurer l'expression des gènes et réaliser des positionnements et contrôles (figure 1.21). Sur les 33 000 spots de la puce, près de 4 000 sont utilisés afin de gérer la qualité : déterminer les problèmes, calculer le signal et le bruit, etc. Ces paramètres permettent d'associer à chaque spot une valeur de confiance. C'est à partir de toutes ces données de contrôle que des prétraitements mathématiques permettent d'obtenir des valeurs d'expression et de confiance. Certains logiciels permettent à l'utilisateur de superviser et d'intervenir dans ces différents processus.

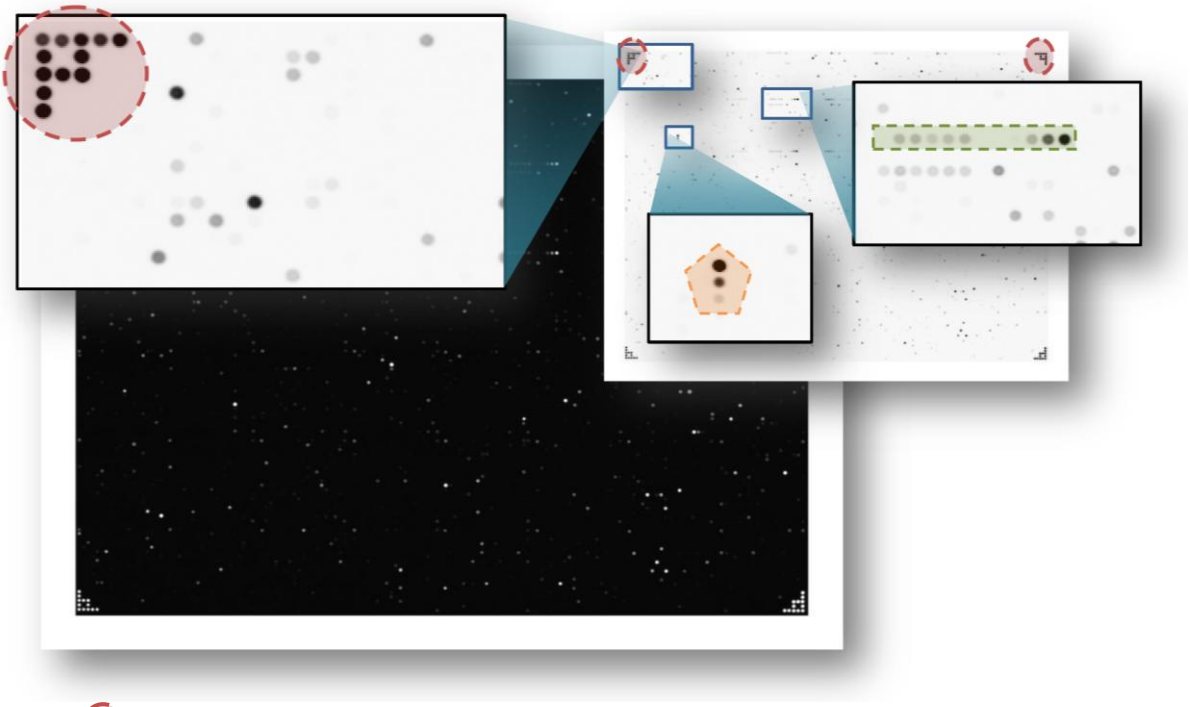


Figure 1.21 – Sur les puces Applied Biosystems, l'expression des gènes est révélée par une chimioluminescence. Pour une meilleure lecture, nous avons inversé la luminescence (noir sur blanc au lieu de blanc sur noir). Sur les 33000 spots de la puce, près de 4000 sont dédiés à l'amélioration de l'analyse d'image et au contrôle de la qualité. Sur cette demi-image d'une puce, par exemple, on peut distinguer très facilement des spots présents dans les coins pour positionner la puce (cercles pointillés rouges). D'autres marqueurs sont présents régulièrement afin d'améliorer la qualité de l'analyse de luminescence, de vérifier que l'hybridation s'est correctement déroulée ainsi que la rétrotranscription (pour laquelle il existe deux kits différents RT et RT-IVT). Enfin, le pentagone orange montre une échelle permettant de calibrer les valeurs de luminescence.

L'analyse des résultats se base sur le fichier produit par le logiciel associé au scanner. Le fichier contient pour chaque gène une valeur d'intensité de chimioluminescence du *spot*, un indice de confiance, un *« drapeau »*[1] permettant de signaler des spots ayant posé des problèmes critiques, un rapport signal bruit, etc. A partir de ces données (33 000 gènes), on souhaite faire ressortir les gènes différentiellement exprimés. Les données ont été manipulées à l'aide de R (un environnement et langage de calcul statistique [Ihaka and Gentleman 1996]), cela se fait aussi souvent au sein d'un tableur. On commence par éliminer les données non fiables : celles dont la valeur du drapeau dépasse 100 ou celles dont le rapport signal/bruit est inférieur à 3. Pour obtenir ces paramètres, une recherche bibliographique des bonnes pratiques de la communauté a été nécessaire. A ce stade, il reste environ 8 000 gènes valides. Le ratio entre l'expression de la puce traitée à l'acide rétinoïque et la puce contrôle est calculé. En appliquant alors une fonction logarithmique, les valeurs positives correspondent à un gène surexprimé en présence d'acide rétinoïque, une valeur négative un gène réprimé. A partir d'un seuil fixé empiriquement, on élimine à nouveau tous les gènes faiblement modulés (dont la valeur absolue du log-ratio est inférieure au seuil). Il reste près d'un millier de gènes. La figure 1.22 présente le résultat graphique de notre expérience.

[1] Traduction du terme anglais *« flag »* plus fréquemment employé.

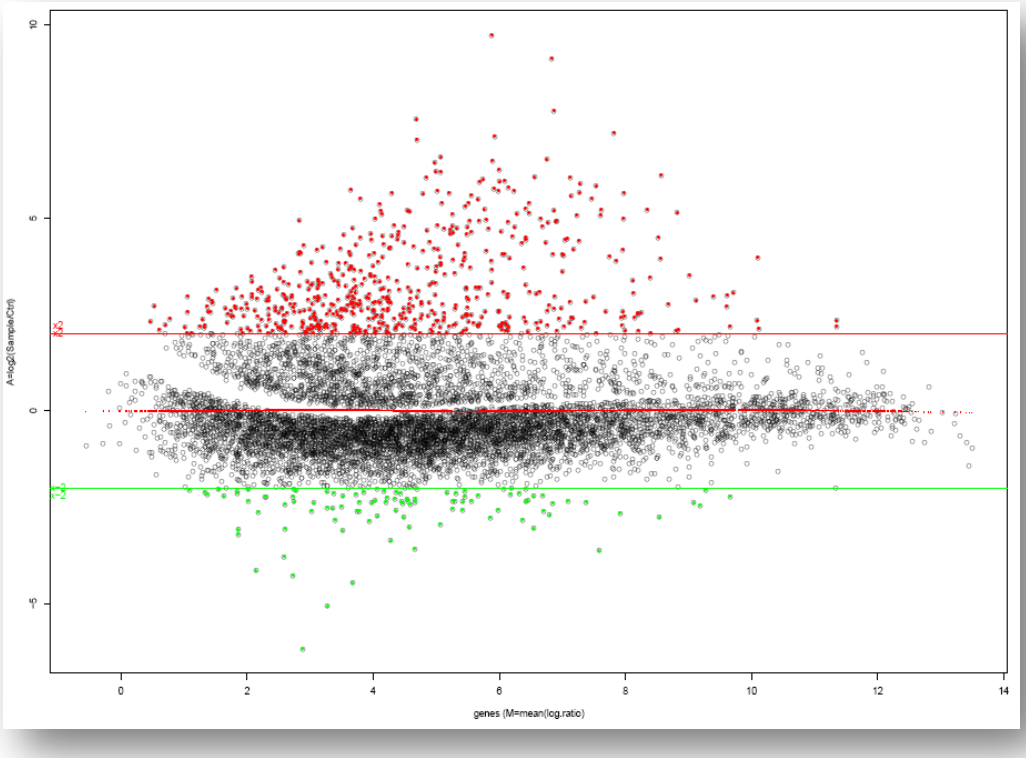


Figure 1.22 – A l'aide d'un outil statistique (ici R), on applique des prétraitements statistiques. Ici l'axe vertical représente le $log_2(traité\ /\ contrôle)$. Les spots rouges au dessus de la ligne haute sont des gènes surexprimés, ils sont plus de quatre fois plus exprimés dans l'échantillon traité que dans celui non traité (contrôle). Les points verts en dessous de la ligne inférieure sont les gènes réprimés. Les gènes intermédiaires (en noir) sont ceux considérés comme insuffisamment modulés. L'abscisse représente pour chaque gène le $log_2((traité + contrôle)/2)$. Plus un gène est à la droite du graphique, plus il est fortement exprimé dans les deux cas. Plus il est à gauche, moins il est exprimé. Sur les 8000 gènes, seule une centaine répond à nos critères.

D'un point de vue expérimental, il est plus facile en biologie d'inhiber un gène que d'augmenter son niveau d'expression. Pour l'inhiber, on diffuse une protéine qui s'apparie à celle produite par le gène. Les protéines produites par le gène sont ainsi bloquées. Pour ces raisons, nous nous focalisons sur les gènes réprimés en présence d'acide rétinoïque. A ce stade, on utilise des logiciels permettant de faire des regroupements, de la classification automatique, et éventuellement d'intégrer des données depuis les portails du domaine. L'application discutée dans le chapitre 7 (section 7.3 page 201) de ce mémoire concerne cette étape de l'exploitation des résultats. Le long travail d'interprétation de ces résultats repose essentiellement sur l'expertise du biologiste.

Actuellement, nous en sommes à cette étape de l'analyse. La suite de l'expérience n'est donc qu'une projection de suppositions. Dans un premier temps, à partir des résultats obtenus, l'expression différentielle des gènes d'intérêt doit être vérifiée individuellement à l'aide de procédures fiables (QRT-PCR). A plus long terme, le biologiste nouerait de nouvelles collaborations avec des équipes spécialisées dans la construction d'un inhibiteur adéquat qu'il faudrait expérimenter. Durant ce temps, un travail de veille scientifique est indispensable. Finalement, les données et connaissances peuvent être publiées et partagées (soumission aux portails du domaine).

Le schéma ci-dessous (figure 1.23) illustre l'omniprésence de l'informatique dans la démarche expérimentable du biologiste. L'informatique est représentée au travers de trois axes (non exhaustifs) : la gestion des données, l'analyse et le traitement des données, et enfin la recherche d'information. La démarche expérimentale est divisée en quatre étapes : la

préparation de l'expérience, sa mise en œuvre, l'analyse de données résultat, et enfin le partage et la publication des résultats. Bien sur, cette démarche, bien qu'ordonnée, est cyclique. En général, de nombreuses expérimentations sont effectuées avant d'arriver à un résultat publiable. Ce sont les résultats d'une expérience qui orientent ainsi la suite de l'étude. Ces découpages peuvent être discutés, raffinés et ne sont pas exhaustifs. Ils reflètent cependant bien le quotidien du biologiste. Signalons enfin l'absence dans la grille et dans ce chapitre plus généralement des aspects pédagogiques. L'enseignement concerne un grand nombre de chercheurs, et motive différentes contributions. Dans les mêmes thématiques que précédemment par exemple, la simulation a donné lieu à la conception d'un laboratoire virtuel [Iazzetti, Santini et al. 1998].

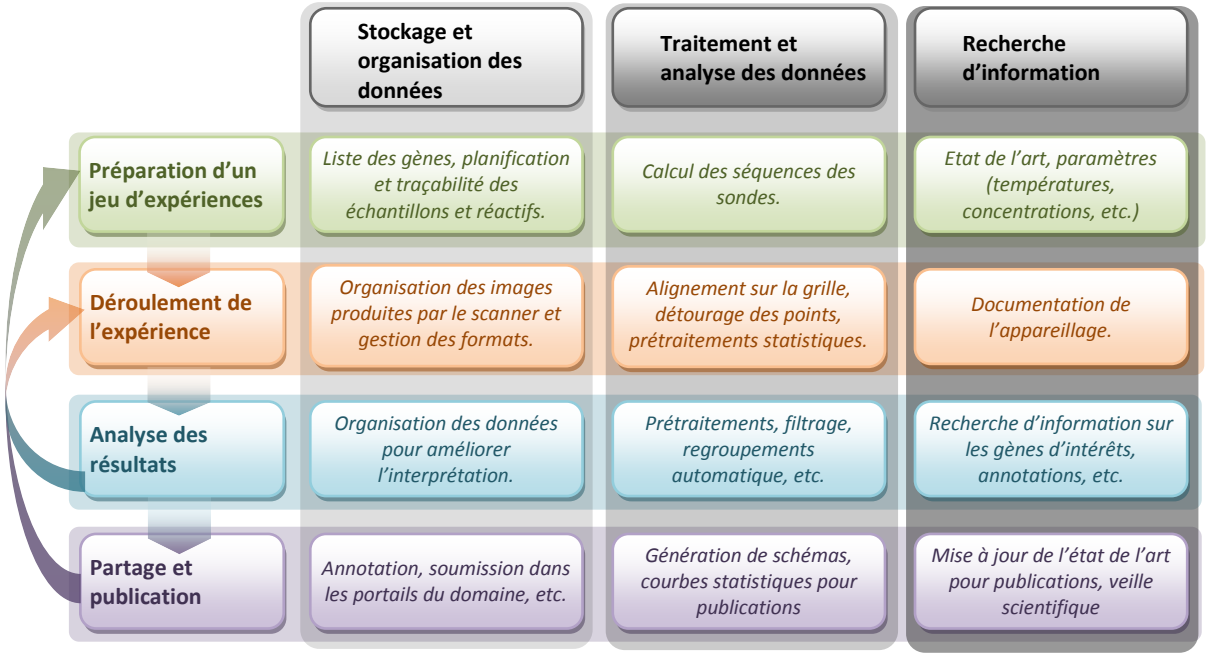


Figure 1.23 – Synthèse illustrant la présence de l'informatique dans une démarche d'analyse par puces à ADN de Plasmodium Falciparum. Un grand nombre d'étapes est commun à d'autres expériences.

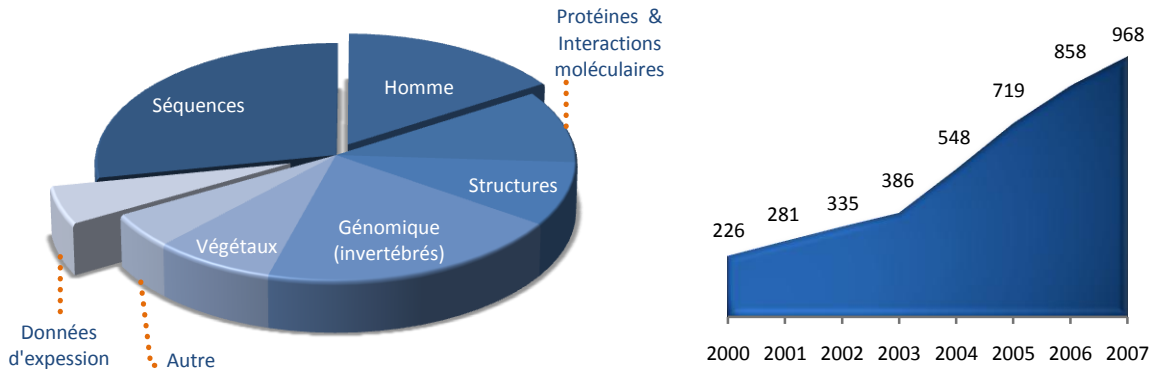


Figure 1.24 – Nucleic Acids Research propose chaque année un numéro de sa revue dédié à l'inventaire des bases de données disponibles pour le biologiste. Entre 2000 et 2007, le nombre de bases recensées est passé de 226 à 968. Le diagramme en « camembert » à gauche montre la répartition de ces ressources. Elle tient compte de la multiplicité des thématiques d'une base de données (il y a environ 20% de plus d'indexation que de nombre de bases de données). On constate que les plus grandes thématiques sont les séquences, la génomique, l'homme enfin les protéines et leurs interactions moléculaires. Les données d'expression représentent aussi une part non négligeable des banques de données, alors même que la proportion de biologiste qui peut financer ce type d'expérimentation est moindre. Ceci montre notamment l'importance de la collaboration entre informatique et biologie dans le contexte de dispositifs haut-débits.

1.3.2 De multiples systèmes d'information pour de multiples besoins

Après avoir présenté la démarche expérimentale dans sa globalité, cette section décrit plus précisément les systèmes d'information. Ils constituent le cœur de la bioinformatique : les bases de données de séquences nucléotidiques sont souvent évoquées pour leur volume important, et le nombre de portails en ligne recensés est tel que la plupart des chercheurs n'en connait qu'un faible nombre. La revue *Nucleic Acids Research* a récemment dénombré 968 systèmes d'information partagés (cf. figure 1.24) [Galperin 2007]. Il est difficile de hiérarchiser ces sites, de les classer d'une façon stricte. Certains sont plus généralistes, d'autres se focalisent sur un organisme. On retrouve généralement le cloisonnement génomique – transcriptomique – protéomique – interactions moléculaires, mais de nombreux autres critères doivent êtres utilisés pour les classer : maladie, pharmaceutique, organisme d'étude, etc. Certains systèmes sont implantés par des industriels comme service associé à leurs produits, afin d'attirer ou de fidéliser le client, tandis que d'autres sont issus de recherche en bioinformatique. Le public visé varie aussi amplement, d'une grande proportion de la communauté à quelques équipes ou dizaines de chercheurs dans le monde.

Les portails suivants sont présentés par leurs fonctionnalités, sans détailler leur architecture, niveau sémantique, disponibilité, etc. Nous avons le plus souvent choisi les exemples en raison de leur utilité pour nos collaborateurs autour des projets concernant *Plasmodium Falciparum* et la leucémie.

Portails bibliographiques

Les domaines de la biologie et du médical ont la particularité d'avoir une excellente structuration des données bibliographiques. En effet, PubMed (anciennement Medline) est un portail dépendant de la NLM (National Library of Medicine) appartenant aux NIH[1][Wheeler, Barrett et al. 2006]. Ce portail, que l'ont pourrait comparer à l'INIST en France a une portée cependant bien supérieure : il recense toutes les publications relatives à la biologie et au biomédical depuis les années 1970. Il référence ainsi actuellement plus de 18 millions d'articles issus de plus de 5000 revues, avec une croissance de 660 000 articles durant la dernière année (figure 1.25 & figure 1.26). Il propose, entre autres, des services avancés de recherche et des mécanismes d'alerte. Enfin, lorsque cela est disponible, il ajoute un lien vers le texte intégral du document.

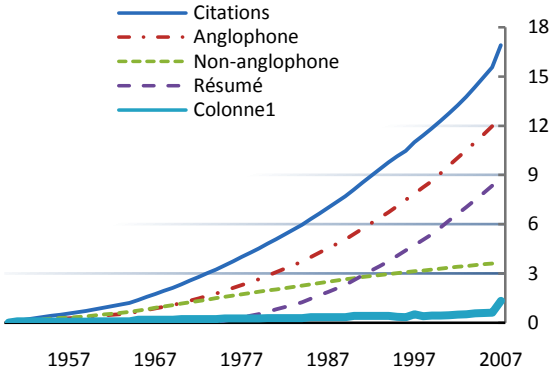


Figure 1.25 – Courbe de croissance du nombre de références enregistrées par PubMed. La courbe totale (trait continu) est adjointe des courbes représentant les proportions de documents rédigés en anglais ou non et de références comprenant des résumés (en anglais uniquement).

Figure 1.26 – Ce graphique montre la croissance du nombre mensuel d'accès au portail entre 1997 et septembre 2006 (en millions). Il est actuellement de l'ordre 75. La ligne représente la moyenne mobile établie sur la période des 12mois qui précède.

[1] Equivalent aux Etats Unis du ministère de la santé

PubMed propose un grand nombre de références, la moitié avec leur résumé (en anglais uniquement). Cependant, de nombreuses revues mettent aussi à disposition leur contenu sur Internet. Certaines sont entièrement payantes, d'autres comme Bioinformatics mettent à disposition le contenu datant de plus d'un an. Enfin, certains portails comme PubMed Central (créé en 2000) ou BioMed Central proposent la totalité de leur contenu en accès gratuit. BioMed Central contient en plus un annuaire des portails et bases de données interrogeables en ligne, et prépare une bibliothèque d'images et vidéos. PubMed central propose quant à lui des contenus supplémentaires liés aux articles.

Ces premiers portails partagent donc les publications et leur référencement. D'autres ressources ne proposent pas des publications, mais des synthèses de publications liées à des domaines particuliers. Wikipédia est alimenté par un grand nombre de contributeurs suivant le concept du Wiki Wiki Web [Cunningham and Leuf 2001]. Ses descriptions restent cependant sommaires au regard des connaissances d'un expert impliqué dans un domaine pendant plusieurs années. OMIM diffuse un contenu plus complet et concerne les affections génétiques humaines. Ce projet est issu d'un ouvrage plus ancien datant des années soixante et ayant atteint sa 12ème et dernière édition en 1998 [McKusick 1998]. Par exemple, la page concernant le gène G-CSF contient 35 citations PubMed, est rédigée par 5 contributeurs et a été révisée 23 fois. Actuellement, OMIM propose plus de 18 300 articles.

Portails généralistes sur les séquences des gènes et protéines

Du séquençage massif de génomes il a résulté une collaboration internationale pour partager les séquences en ligne. L'*« International Nucleotide Sequence Database Collaboration »* réunit trois organisations : le NCBI aux Etats-Unis, EMBL en Europe (qui dépend de l'EBI), et DDBJ au Japon). Chaque miroir partage les génomes de plus de 165 000 organismes pour un total de 100 milliards de nucléotides[1] (cf. figure 1.27). La base de données RefSeq propose des séquences de référence, c'est-à-dire plus fiables car validées par un expert [Pruitt, Tatusova et al. 2005]. Concernant les séquences de protéine, un consortium s'est aussi fondé, UniProt, qui regroupe SwissProt du SIB, TrEMBL de l'EBI, et PSD de PIR. L'ensemble de ses données les plus fiables, UniProtKB, contient 4,25 millions d'entrées (non redondantes) [Apweiler, Bairoch et al. 2004; Consortium 2007]. PDB est une base de données fortement utilisée, extérieure au consortium d'UniProt, dont la taille est plus restreinte mais qui contient des informations structurales plus complètes [Kouranov, Xie et al. 2006].

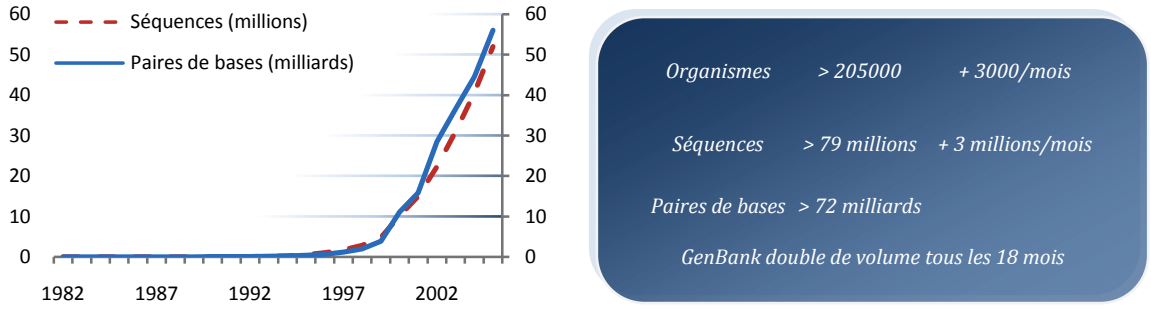


Figure 1.27 – Croissance des données nucléotidiques de GenBank (à gauche) et statistiques actuelles (à droite).

Interactions moléculaires

Alors que la génomique, la transcriptomique et la protéomique relèvent des portails précédents, la connaissance en aval, qui concerne la biochimie ou qui s'intéresse à toutes les interactions moléculaires, se situe dans d'autres portails. Le plus connu est KEGG (Kyoto Encyclopedia of Genes and Genomes), qui a débuté par les voies métaboliques mais qui

[1] http://www.nlm.nih.gov/news/press_releases/dna_rna_100_gig.html

actuellement partage une connaissance sur les médicaments, voies de signalisation, etc. Les voies métaboliques étaient représentées bien avant l'existence de ce portail dans de grands schémas indiquant les réactions moléculaires à différents niveaux. KEGG s'apparente à un grand graphe découpé en schémas, chacun représentant une voie métabolique. Chacun de ces schémas est un sous-graphe dont les arêtes représentent les réactions chimiques étiquetées par le ligand, reliant à l'origine un substrat et pointant vers le produit de la réaction. Certains schémas encapsulent des informations complémentaires comme la position des membranes cytoplasmiques et nucléaires, des compartiments cellulaires, etc. Pour certains biochimistes, ces graphes sont de véritables *cartes* métaboliques dont le formalisme et la disposition sont fo .28).

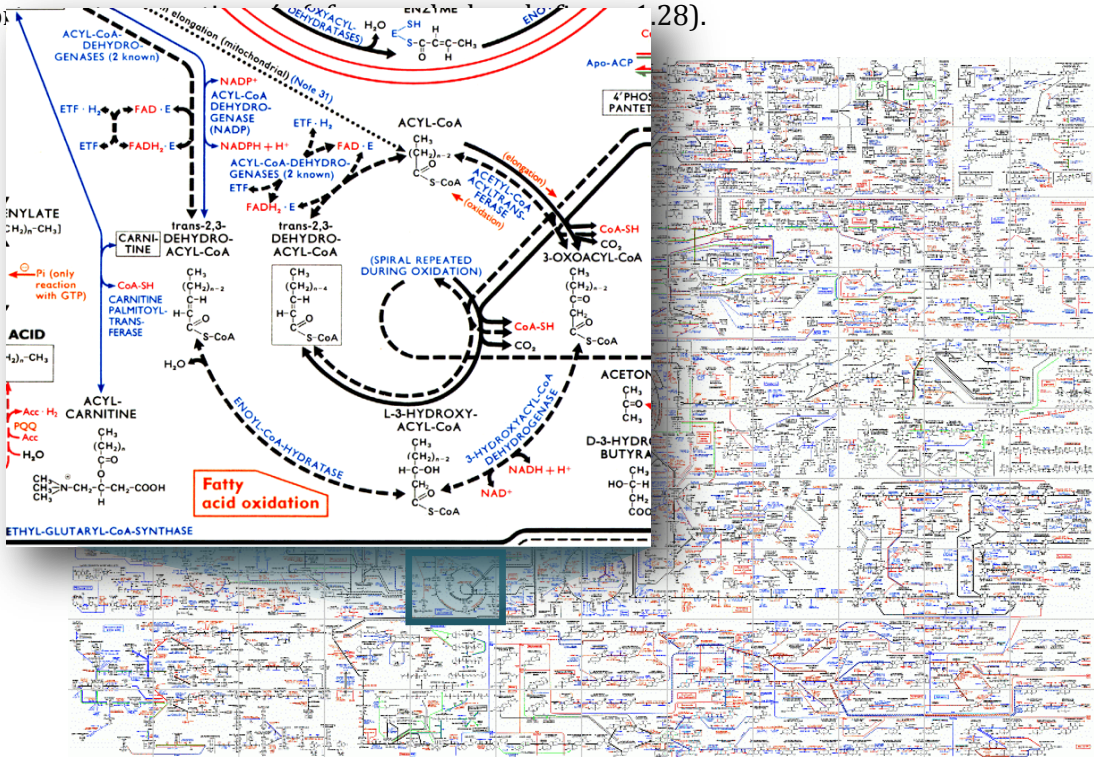


Figure 1.28 – Les voies métaboliques constituent de gigantesques cartes dont la représentation est conventionnelle. La carte ci-dessus est dessinée manuellement et diffusée sous forme de poster. D'un point de vue topologique, il s'agit d'un grand hypergraphe reliant substrat et ligand au produit d'une réaction chimique. Cette figure est mise à disposition par ExPASy[1].

Dans KEGG, cette description d'interactions moléculaires est disponible pour de nombreux organismes. Une réaction chimique peut être étiquetée par plusieurs ligands. Ceci signifie que chaque ligand peut provoquer la réaction. Lorsque l'on choisi un organisme, tous les ligands connus pour un organisme sont alors teints en vert. On peut rapidement distinguer quelques chemins métaboliques identifiés par la communauté pour l'organisme concerné (cf. figure 1.29). D'autres liens relient aussi les voies métaboliques entre elles et permettent de naviguer de sous-graphe en sous-graphe. Enfin, comme son nom l'indique, KEGG propose des liens interactifs pour chaque élément du schéma permettant d'ouvrir une page avec une description précise de la molécule mise en jeu : nom, séquence protéique, annotations, lien vers les portails les plus courants, etc. D'un point de vue technique, les données ont été initialement saisies manuellement par les opérateurs participant au projet. La nomenclature utilisée pour les enzymes est l'EC (Enzyme Classification).

KEGG s'adresse à un public assez large de biologistes et de biochimistes. Cette représentation se situe à un très bas niveau au regard d'une grande partie de la communauté. D'autres portails proposent des services similaires. Reactome.org est un projet européen généraliste. De

[1] http://www.expasy.ch/cgi-bin/show_thumbnails.pl

nombreux portails relatifs à des domaines spécifiques proposent des versions nettoyées et corrigées (*« curated »*) par rapport au domaine. Nos contacts au CEA par exemple n'utilisaient pas KEGG mais naviguaient dans les voies métaboliques proposées par le portail TAIR spécifique à l'*Arabidopsis Thaliana* [Rhee, Beavis et al. 2003].

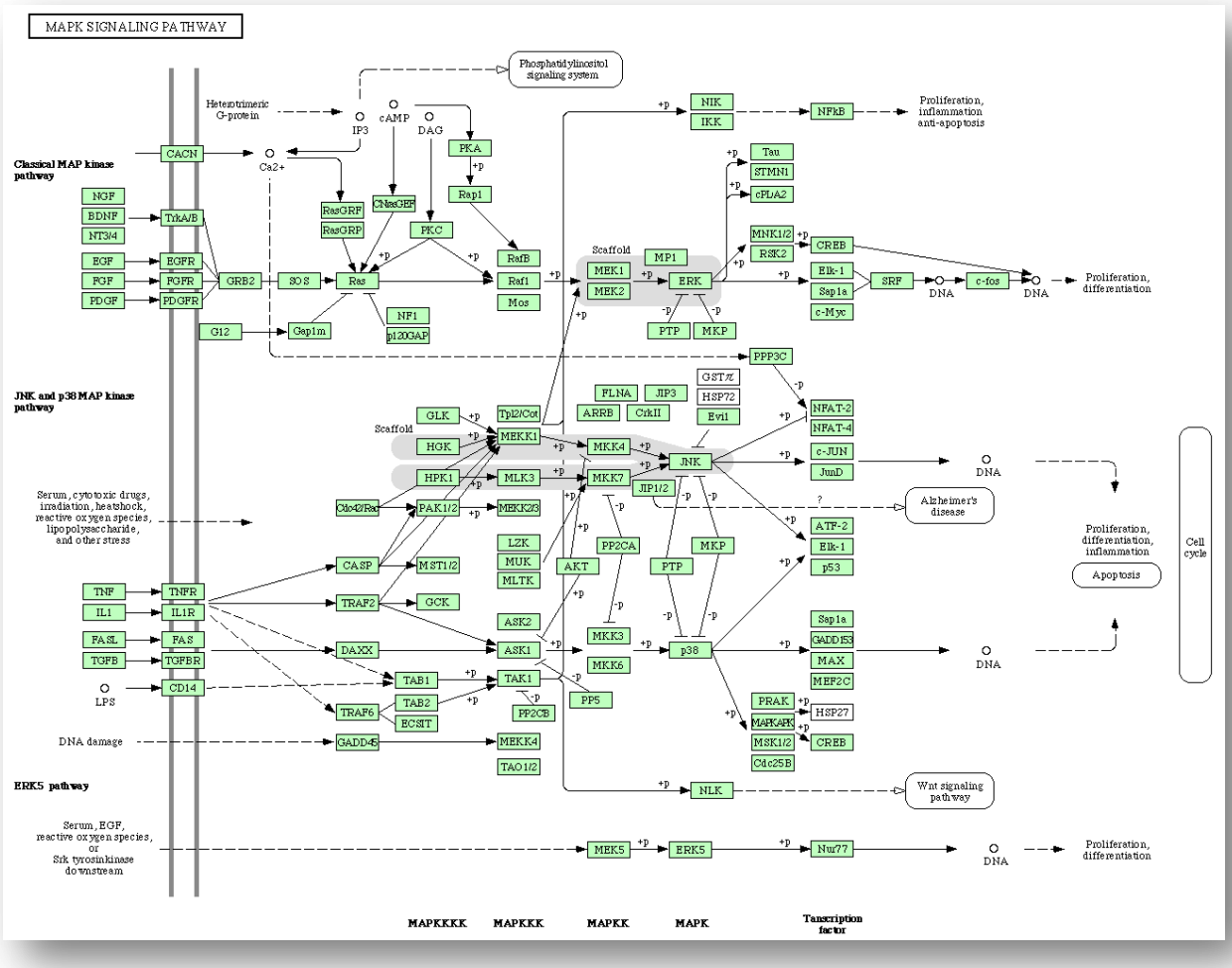


*Figure 1.29 – KEGG partage des schémas d'interactions moléculaires. Il s'agit ici d'une voie de signalisation qui concerne la protéine MAPK (*Mitogen-Activated Protein Kinases*) impliquée notamment dans les mécanismes de prolifération*[1]*. Les éléments en vert sont constatés chez l'homme. La double ligne verticale sur la gauche représente la membrane de la cellule. Les boites de textes arrondies blanches sont des liens vers d'autres voies métaboliques de KEGG.*

Nomenclature et normalisation

Nous avons mentionné à plusieurs reprises la présence de standards de nomenclature : MeSH comme vocabulaire contrôlé de PubMed, ou encore EC pour les enzymes. Nous reviendrons sur des définitions plus précises dans le prochain chapitre ; considérons simplement pour l'instant qu'il s'agit d'un ensemble de termes utilisés comme convention au sein d'une communauté pour un usage automatisé, pour éviter les problèmes liés à la synonymie, etc. Il existe de nombreuses ressources visant à normaliser les usages dans les sciences du vivant. On peut regrouper ces ressources normalisatrices autour de trois axes :

- la terminologie du domaine

[1] http://en.wikipedia.org/wiki/MAPK

- les noms des gènes, protéines enzymes (que l'on généralise par le terme entité nommée)
- les identifiants numériques (ou numéros d'accession).

La définition de ces ressources sera plus amplement abordée dans le chapitre suivant, d'ici là, nous les appelons terminologies. Citons quelques ressources parmi les plus répandues. En matière de terminologie, outre le MeSH, Gene Ontology est un ensemble de trois terminologies utilisées pour annoter les gènes [Ashburner, Ball et al. 2000; Consortium 2006]. Elle permet de caractériser le processus biologique, la composante cellulaire et la fonction moléculaire des gènes ainsi que des transcrits et des protéines qui en découlent. Entrez Taxonomy est une hiérarchie des espèces sensée refléter l'évolution [Benson, Karsch-Mizrachi et al. 2006; Wheeler, Barrett et al. 2006]. Enfin, citons l'initiative d'UMLS, un entrepôt visant à rassembler plus d'une centaine de terminologies et de les rendre interopérables [Bodenreider 2004]. Des préoccupations voisines existent en France : l'INIST est connu pour son portail documentaire scientifique national. Il a récemment mis en œuvre un projet du nom de TermSciences qui encapsule plusieurs terminologies de domaines scientifiques différents. Notons qu'une de ces ressources est la traduction française du MeSH réalisée par l'INSERM.

De nombreux portails utilisent ces différentes ontologies ; GO est en particulier la ressource essentielle utilisée pour l'annotation. Elle est ainsi employée par les nombreux portails cités jusqu'ici (GenBank, RefSeq, UniProt, PDB, TAIR[1], etc.), mais aussi dans des portails dédiés comme GeneDB, GOA, etc. GeneDB est un portail qui rassemble les données des séquençages réalisés par la *« Pathogen Sequencing Unit »* du *« Wellcome Trust Sanger Institute » [Hertz-Fowler, Peacock et al. 2004].* Ce système d'information a de particulier qu'il est la référence utilisée par le portail PlasmoDB relatif au génome de *Plasmodium Falciparum*. GOA (Gene Ontology Annotation) est un projet mené par l'EBI et visant à regrouper les annotations mises à disposition par de nombreux contributeurs, dont GeneDB, HGNC, Ensembl, Reactome, TAIR, TIGR, etc.[2] [Camon, Barrell et al. 2003; Camon, Magrane et al. 2004]. GOA est proposé sous forme de plusieurs distributions[3] (cf. figure 1.30). La plupart des portails lient des annotations de GO à leurs produits de gènes, et lorsque ce n'est pas le cas, les références vers des portails du domaine permettent d'y accéder indirectement et rapidement.

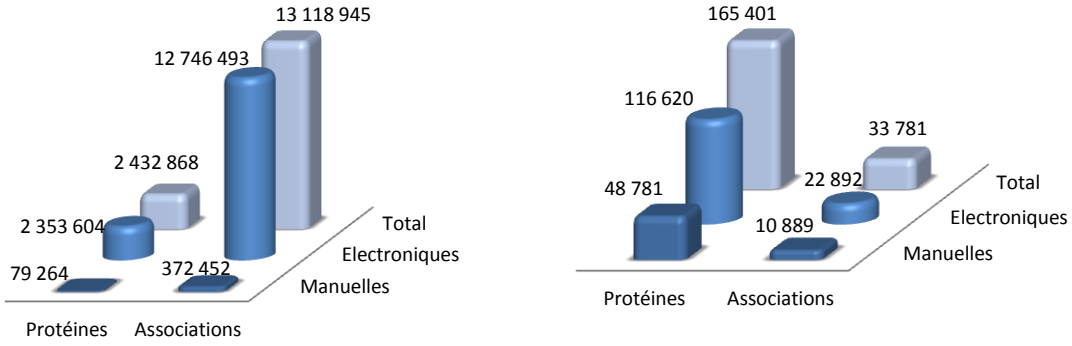


Figure 1.30 – GOA propose notamment des distributions relatives à l'humain ou à UniProtKB. On constate qu'UniProt contient une plus grande quantité de données, mais une très faible proportion résulte d'une expertise (≈ 3%), le reste ayant été généré par des prédictions basées sur les séquences et autres méthodes automatiques. Au contraire, chez l'homme, près d'un tiers des données sont fiables et ne sont pas issues de procédures automatiques.

[1] En réalité, TAIR repose sur des ontologies qui ont été intégrées dans GO.

[2] Pour une liste exhaustive des contributeurs, *confer* `http://www.ebi.ac.uk/GOA/goaHelp.html`

[3] sous-ensembles de données

Concernant les numéros d'accession, de nombreux grands portails de référence s'imposent et leurs identifiants numériques sont utilisés pour interopérer. C'est le cas par exemple du « GI » (GenBank Identifier) qui est utilisé par une grande partie de la communauté pour identifier les séquences. Mais aucune réelle standardisation n'est imposée ou régulée par une autorité, hormis concernant la publication scientifique pour laquelle PubMed est incontournable. Pour permettre de rendre ces systèmes d'information interopérables, certains portails permettent le référencement croisé entre les différentes ressources et proposent des fichiers d'alignement. C'est le cas exemplaire d'Entrez Gene (anciennement Locus Link) [Pruitt and Maglott 2001; Maglott, Ostell et al. 2005]. Pour l'utilisateur, de nombreux portails proposent des liens directs vers d'autres portails, de référence ou du domaine, en indiquant éventuellement les identifiants correspondant (cf. figure 3.4 page 85).

Enfin, si les entités nommées sont rarement structurées lors de leur création, elles le sont de plus en plus a posteriori. En effet, héritage de l'histoire, les noms des gènes et protéines relevaient souvent d'une étymologie hasardeuse (initiales du chercheur ayant séquencé, etc.) : les entités nommées émergeaient sans contrôle. C'est pour cela qu'elles sont difficiles à manipuler aujourd'hui : problème de synonymie, casse aléatoirement importante, etc. Les systèmes d'information privilégient donc une communication structurée par les identifiants numériques de référence. Cependant, les chercheurs durant la rédaction d'articles préfèrent les noms des gènes, tout comme certains fabricants de puces par exemple. La reconnaissance des entités nommées est essentielle à la recherche d'information et à la fouille de données, tout comme à l'analyse des données issues de certains dispositifs haut-débits. Actuellement, Genew est la seule initiative importante [Wain, Lush et al. 2002; Eyre, Ducluzeau et al. 2006]. Elle est relative au génome et protéome humain, régie par le HGNC (HUGO Gene Nomenclature Committee) qui dépend de l'HUGO (Human Genome Organization) dans lequel sont impliqués partenaires industriels, universitaires et institutionnels. Pour les autres organismes vivants, ce sont en fait les portails de référencement croisés tels que Entrez Gene qui de la même façon gèrent les noms des gènes et de leur produits.

Portails spécifiques à *Plasmodium Falciparum*

La communauté réunie autour de *Plasmodium Falciparum* est restreinte au regard d'autres espèces comme l'homme. Il existe pour ce parasite quelques ressources, cependant les chercheurs de l'institut Pasteur n'utilisent qu'un seul portail dédié à cet organisme, PlasmoDB, le portail officiel du consortium de séquençage du génome de ce parasite [Bahl, Brunk et al.; The Plasmodium Genome Database Collaborative 2001]. Ce portail est construit à partir du schéma de GUS (Genomics Unified Schema) qui contient près de 300 relations et permet d'intégrer les données génomiques issues des principaux projets (séquences, annotations, etc.) [Davidson, Crabtree et al. 2001]. D'autres ressources concernent cet organisme et proposent une distribution de leurs données restreinte à cet organisme. C'est le cas de certaines ressources généralistes comme GenBank, RefSeq, Entrez Gene, UniProt, KEGG ou d'autres ressources ayant des thématiques spécifiques mais transversales à plusieurs organismes : par exemple MPIM (*« Mitochondrial Protein Import Machinery »)* [Lister, Murcha et al. 2003] se focalise sur les mécanismes énergétiques. Full-Malaria partage les séquences complètes d'ADNc[1]. D'autres portails regroupent les *apicomplexes* (terme qui regroupe les genres *Plasmodium*, *Cryptosporidium* et *Toxoplasma*) (ApiEST-DB [Li, Crabtree et al. 2004], ApiDB [Aurrecoechea, Heiges et al. 2007], Comparasite [Watanabe, Wakaguri et al. 2007]).

Portails spécifiques au projet sur la cancérologie promyélocytaire aiguë

Comme nous l'avons déjà abordé, l'homme est l'un des organismes qui motive le plus grand nombre de travaux. *Nucleic Acids Research* dénombre 16 des ressources spécifiques à l'homme

[1] L'ARN est présent en brins assez courts, et se dégrade rapidement. Pour ces raisons on le rétrotranscrit en ADNc (ADN complémentaire). Les chaînes sont dites complètes lorsqu'elles sont reconstituées à partir de fragments. Ces séquences sont notamment utilisées afin de déterminer et synthétiser des protéines.

(génomique, affections et immunogénétique). A cela il faut ajouter les multiples portails généralistes ou pluri-espèces. Le contexte est l'étude de la leucémie à l'aide de puces à ADN. Dans ce contexte, ce sont plus de 300 systèmes d'informations qui sont recensés. Le biologiste est confronté à un grand nombre de ressources, souvent redondantes ou peu utiles. Dans la pratique, la justification d'une si large offre de portails est de fournir à l'utilisateur un outil adapté à son besoin et de lui éviter de recourir à un grand nombre d'outils disparates.

Dans le contexte de notre collaboration avec Y. Cayre, plusieurs besoins sont exprimés. Du point de vue du dispositif expérimental, nous utilisons des puces à ADN produites par Applied Biosystems. L'analyse de données de puces repose dans un premier temps sur les données mises à disposition par le fabricant au sein du portail Panther. Par la suite, il est nécessaire de croiser l'information résultant de nos puces avec les résultats de la communauté. Sont alors concernées :

- les ressources bibliographiques (PubMed et OMIM),
- les ressources qui apportent une information fonctionnelle : en priorité les portails dédiés à l'homme, la cancérologie ou la leucémie (Genew semble la plus pertinente), mais aussi les portails plus généraux (Entrez Gene, GOA, UniProt, KEGG par exemple),
- les ressources liées aux données d'expression permettant de comparer nos données à des expériences similaires (ArrayExpress de l'EBI [Parkinson, Kapushesky et al. 2007], CGED [Kato, Yamashita et al. 2005], et les standards proposés par le MGED [Ball, Brazma et al. 2004; Ball and Brazma 2006]).

1.4 Synthèse

Au vue de ce que nous avons exposé, l'informatique est un outil indispensable pour beaucoup de biologistes. Il se restreint parfois à la communication et à la recherche d'information dans PubMed, socle commun à tous les chercheurs. Mais bien généralement, des besoins spécifiques se font ressentir : l'appareillage nécessite un traitement du signal, il faut commander des kits ou calculer des séquences de sondes, il faut convertir des identifiants d'accession dans des formats standards, rechercher l'information fonctionnelle dans les bases de données du domaine, etc. Les problématiques bioinformatiques sont impliquées au premier plan dans le support aux dispositifs expérimentaux à haut débit. Cependant, de nombreuses techniques plus discrètes que les puces à ADN, productrices de données à plus petite échelle expriment aussi des besoins, et concernent parfois un bien plus grand nombre d'utilisateurs, car moins coûteuses que les dispositifs haut-débit.

Nous avons présenté cette omniprésence de l'informatique en nous appuyant sur les collaborations que nous avons, et en illustrant sous forme d'un *scenario* différents besoins. Cependant, cette description ne peut pas être considérée comme étant représentative de tous les besoins et contextes qui existent dans la recherche en biologie.

Nous avons vu que les données à traiter sont de natures diverses, et si nombreuses que seuls les outils informatisés peuvent les exploiter efficacement. Le chapitre suivant présente l'ingénierie des connaissances et l'intégration de données. Ces deux domaines nous concernent pour stocker, partager et traiter efficacement l'information biologique qui est hétérogène et distribuée.

CHAPITRE 2

Prérequis informatiques et définitions

«During the Middle Ages and early Renaissance, Italy was fragmented into dozens of rival city-states controlled by such legendary families as the Estes, Viscontis and Medicis. Though picturesque, this political fragmentation was ultimately damaging to science and commerce because of the lack of standardization in everything from weights and measures to the tax code to the currency to the very dialects people spoke. A fragmented and technologically weak society was vulnerable to conquest, and from the seventeenth to the nineteenth centuries Italy was dominated by invading powers.

The old city-states of Italy are an apt metaphor for bioinformatics today. The field is dominated by rival groups, each promoting its web sites, services and data formats. Unarguably, this environment of creative chaos has greatly enriched the field. But it has also created a significant hindrance to researchers wishing to exploit the wealth of genome data to its fullest.

Despite its shaky beginning, the nation of Italy was eventually forged through a combination of violent and diplomatic efforts. It is now a strong and stable component of a larger economic unit, the European Union, with which it shares a common currency, a common set of weights and measures, and a common set of rules for national and international commerce. My hope is that bioinformatics will one day achieve the same degree of strength and stability by adopting a universal code of conduct along the lines I propose here.

[…]

The risk, of course, is that like the mediaeval Italian city-states, each of these projects will endorse its own idea of standardization, and a chaotic world of incompatible bioinformatics data standards will be replaced by a chaotic world of incompatible web-service standards. We can look forward to a bit of a struggle before one set of standards achieves pre-eminence, but I have no doubt that unity will be reached eventually. »

LINCOLN STEIN

2.1 Introduction

Le chapitre précédent a introduit des notions élémentaires de biologie moléculaire et présenté une vision de la bioinformatique afin de permettre à un informaticien, non familier du domaine, de cerner le contexte applicatif de notre étude. De la même façon, ce chapitre se veut une synthèse des principales notions informatiques qui nous seront utiles. Il définit le vocabulaire informatique qui sera employé par la suite, et introduit quelques notions et définitions dans trois domaines : l'architecture des systèmes d'information, l'ingénierie des connaissances et la linguistique, et enfin l'intégration de données.

Pour une information plus détaillée et des définitions plus poussées concernant l'origine des bases de données, les relations sémantiques, etc., le lecteur pourra se référer à l'annexe B. Cette annexe introduit de façon détaillée les notions de système d'information, bases de données et d'architecture logicielles. Elle présente le domaine de la linguistique et de l'ingénierie des connaissances avec plus de profondeur. Enfin, elle dresse un inventaire plus complet des systèmes d'intégration existants.

Tout au long de ce chapitre, les notions informatiques abordées seront accompagnées d'exemples liés au domaine bioinformatique et biomédical. Plus spécifiquement, ces exemples seront contextualisés par rapport à nos deux principaux projets qui concernent l'étude de Plasmodium Falciparum et la leucémie promyélocytaire (chez l'homme).

Nous débutons ce chapitre en définissant les termes *« donnée »*, *« information »* et *« connaissance »*. Puis nous nous intéressons à une vue génie logicielle de la problématique. Nous voyons ensuite comment linguistique et ingénierie des connaissances exploitent la sémantique dans les applications. Enfin, nous détaillons la problématique de l'intégration de données et ses caractéristiques.

2.2 Représentation des connaissances

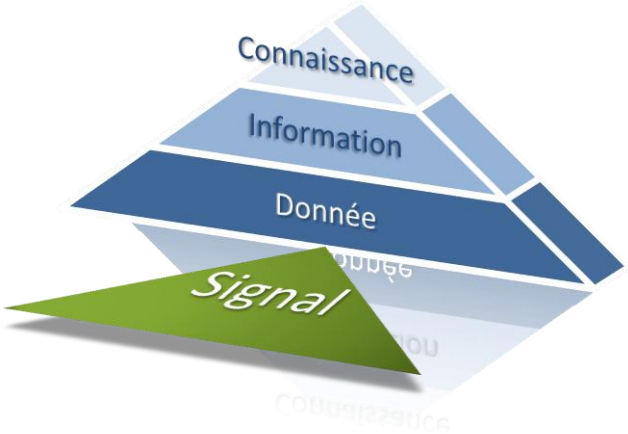


Figure 2.1 – Le signal, non persistant devient une donnée lorsqu'il est codifié, persistant et indépendant de l'appareillage qui l'a produit. L'information est une donnée porteuse de sens et rattachée à un contexte. La connaissance contient l'information mise en forme, mais aussi toute l'information tacite permettant son exploitation.

2.2.1 Données, information, connaissance

Les termes *« donnée »*, *« information »* et *« connaissance »* sont fréquemment utilisés dans de multiples contextes, avec des significations nuancées. Différentes définitions existent ; nous en proposons une synthèse rapide dont le point de départ est la notion de **signal** (figure 2.1) : il est

physique, le plus souvent électrique, magnétique ou ondulatoire, et permet la communication entre plusieurs appareillages. Il est non persistant et dépendant du média qui le transporte. La **donnée** est le stockage (persistant) du signal éventuellement combiné avec un système de décodage. Elle est ainsi exploitable indépendamment de l'appareillage qui l'a générée. Parmi les définitions de référence, voici celle de l'AFNOR :

> *La **donnée** est l'enregistrement dans un code convenu d'une observation, d'un objet ou d'un phénomène (donnée « factuelle ») d'une image, d'un son, d'un texte. C'est un fait, une notion, une instruction représentée sous une forme conventionnelle, convenant à une communication, une interprétation ou un traitement soit par l'homme, soit par des moyens informatiques. (Afnor)*

La donnée est factuelle ; elle devient **information** dès lors qu'elle est mise en contexte et porteuse de sens. Serge Miranda la définit ainsi :

> ***L'information** est alors tout le signifiant que l'on attache et que l'on peut déduire d'un ensemble de données, de certaines associations entre données. [Miranda 2002]*

Le document contient des informations. Par ailleurs, *« assimiler Information et Connaissance est une approximation lourde de confusion […] le terme connaissance étant beaucoup plus vaste que celui d'information, qui ne réunit que l'ensemble des connaissances mises en forme, c'est à dire explicitées »* [Maniez 2005]. Le terme *« **connaissance** »* est l'objet de nombreuses définitions provenant de diverses disciplines : philosophie, psychologie, sciences de l'information, sciences humaines et sociales, etc. Nous avons retenu la définition d'Audrey Mazuy :

> *« La connaissance renvoie à la capacité de disposer d'une représentation mentale d'une réalité plus ou moins bien circonscrite… Toute connaissance d'un objet au sens le plus large du terme implique ainsi de disposer de descripteurs, de valeurs et de relations, et va dans le sens d'une théorisation, qui tend à être partagée, soit par un groupe social, soit par la société toute entière. » (Audrey Mazuy)*[1]

L'information est alors une connaissance explicite et persistante et objective [Maniez 2005] : quels que soient les lecteurs, une même ressource diffuse une même information à différents lecteurs. La connaissance est définie par les philosophes comme un *acte de pensée* ; elle est liée à une représentation mentale, acquises par l'expérience dans le monde réel duquel on ne peut dissocier une dimension sociale. La connaissance est partagée, avec une relative nuance, distorsion et subjectivité.

D'un point de vue plus formel, on peut considérer la connaissance comme un savoir muni d'un ensemble de règles et d'opérateurs permettant le traitement d'une information dans un contexte d'utilisation bien déterminé.

Dans certaines applications informatisées, dans la lignée des recherches en intelligence artificielle, le but consiste à extraire l'information à partir de données (souvent textuelles) pour inférer de *la connaissance*. Pour cela, une analyse poussée est nécessaire concernant les termes, les concepts et les relations sémantiques qui les lient : les ressources termino-ontologiques.

2.2.2 Mot, terme et concept

Avant de présenter les ressources termino-ontologiques, il est nécessaire de descendre à un niveau de granularité inférieur et de présenter les éléments atomiques qui constituent ces ressources, en commençant par le « mot ».

[1] http://audreymazuy.free.fr/dissert/intro.html

*« Le **mot** est une suite de caractères graphiques formant une unité sémantique et pouvant être distinguée par un séparateur (blanc typographique à l'écrit, pause à l'oral). » [Schwab 2005]*[1].

Cette définition simple pose de multiples ambiguïtés d'un point de vue linguistique et ne représente qu'une unité syntaxique. On emploie alors des termes plus spécifiques comme le *lemme* (ou *lexie*), et la *forme fléchie* (ou *flexion*). Le **lemme** se définit intuitivement comme l'entrée d'un dictionnaire. Par exemple, *« généraliste (adj.) »*, *« généraliste (n.m.) »*, *« général (n.m.) »* et *« général (adj.) »* sont quatre lemmes différents. On distingue l'adjectif et le nom qui s'écrivent de la même façon, on ignore cependant le problème de la polysémie (multiplicité des sens d'un mot). Par exemple, le nom *« généraliste »* désigne une entité qui n'a pas de spécialisation ou un praticien de la « médecine générale ». Cette deuxième discipline médicale étant considérée comme une spécialité. Un lemme peut avoir plusieurs **formes fléchies** (conjugaison d'un verbe, genre et nombre pour des noms, adjectifs, pronoms, etc.). Par exemple : *« général (adj.) »* possède plusieurs flexions : *« général »*, *« générale »*, *« générales »*, *« généraux »*.

D'après François Rastier, le mot devient **terme** par une procédure de normalisation : *« le terme est un artefact de la discipline qui l'instaure. […] Le terme est une unité factice de médiation entre la pensée rationnelle et le langage. »* [Rastier 1995]. La pensée fait alors référence au **concept**. Une conception de la relation entre terme et concept considère que le concept préexiste au terme. Le terme est le signifiant du concept qui est le signifié : *« le terminologue normalise l'expression des concepts du domaine en fixant les termes qui le désignent » [Zweigenbaum 1999]*. Le concept est indépendant de la langue.

La perception du monde est simultanément individuelle et partagée. Rastier définit ainsi les concepts comme « *des signifiés normés par les disciplines »*, du point de vue de la Terminologie[2] [Rastier 1995]. Il ajoute que les concepts ne préexistent pas à la langue, ils sont le produit de l'instauration des termes. Il y a donc une indépendance linguistique voulue entre le concept et le terme ; mais ce concept n'est en aucun cas à la source du terme.

La dernière notion présentée dans cette section est celle **d'entité nommée**. Le terme et l'entité nommée sont tous deux employés dans la communication humaine, dans le langage naturel avec une granularité de même niveau. Ils sont cependant distincts d'un point de vue théorique : l'entité nommée n'est généralement pas considérée comme un terme car elle n'est pas paraphrasable. Dans la pratique, les entités nommées sont aussi très nombreuses ; leur registre évolue bien plus rapidement que celui du vocabulaire commun de la langue. Ainsi en quelques années, ce sont des millions de séquences de gènes, protéines et éléments qui ont été nommés, par l'usage et non au travers d'une autorité ou en suivant une règle pour l'essentiel. Dans la pratique, les entités nommées peuvent contenir des virgules, parenthèses, et multiples caractères spéciaux qui ne forment pas de mot dans les langues indo-européennes. Enfin, il n'existe pas réellement de dictionnaire qui inventorie exhaustivement ces entités nommées.

2.2.3 Relations thématiques et sémantiques

Ces entités ayant été définies, leur articulation dans les ressources termino-ontologiques doit être analysée. C'est ce que nous allons voir dans l'étude des différentes relations qui permettent leur structuration.

[1] Didier Schwab restreint cette définition aux langues indo-européennes écrites. Dans le cadre de la bioinformatique, cette restriction n'est pas problématique.

[2] Nous distinguons la « Terminologie » comme la discipline s'intéressant à la conception de ressources terminologiques, des « terminologies » (sans majuscules) qui en sont les produits.

2.2.3.1 Définitions générales sur les relations

Connexionnisme, relation thématique et analyses distributionnelles

Le connexionnisme est un courant des sciences cognitives et de la psycholinguistique qui considère les phénomènes mentaux et comportementaux comme des *« processus émergents d'un réseau d'unités simples interconnectées »* [Wikipedia]). Cette hypothèse est fondée sur plusieurs constats à différents niveaux de granularité. Au niveau cellulaire, la biologie a par le passé décrit le fonctionnement du cerveau par ses neurones. A un plus haut niveau, Alan Collins et Ross Quillian ont observé à la fin des années 60 que le temps d'évaluation de la validité de la proposition *« un chien est un mammifère »* était plus long que celui de la proposition *« un chien est un animal »* [Collins and Quillian 1969]. Cette différence serait liée à la fréquence d'usage [Landauer and Freedman 1968; Collins and Quillian 1970; Juola and Atkinson 1971].

Les relations thématiques découlent essentiellement de cette approche. Elles ont pour objectif de représenter l'association (ou non) entre différentes idées et les thématiques qu'elles partagent. On utilise ainsi le tri par carte par exemple pour organiser des menus, ou plus généralement pour trier des informations [Tullis 1985]. Des thésaurus sont employés pour indexer une information [Roget 1852; Pechoin 1999], etc. Les réseaux sémantiques ont pour objet d'acquérir ces relations thématiques à partir d'analyses linguistiques distributionnelles (cooccurrences, information mutuelle, etc.) [Harris, Gottfried et al. 1989] [Church and Hanks 1991], cf. [Manning and Schütze 1999] pour une revue complète.

La relation thématique est souvent non orientée et ne peut être employée dans un raisonnement. Elle est le plus souvent issue d'approches distributionnelles et probabilistes. Elle permet de construire des réseaux thématiques qui ont un intérêt, mais à l'échelle individuelle, elles ne sont pas significatives.

Relation sémantique

Les relations thématiques n'établissent généralement pas de liens formels entre des concepts, et sont liées à la langue et à son usage. Les relations sémantiques se veulent modélisatrices ; elles visent à formaliser une sémantique entre des concepts exploitable dans le cadre d'un raisonnement. Elles sont le plus souvent orientées (mise à part la synonymie).

Les relations sémantiques sont présentes dans tous les niveaux énumérés (cf. 2.2.2 page 44): niveau lexical (fonctions lexicales de Mel'čuk [Mel'čuk 1988]), terminologique (termes) et conceptuel (concepts). Elles sont généralement liées à des calculs formels (logiques, grammaires, etc.). [Gómez, Moreno et al. 2000] vont plus loin et affirment que la relation sémantique est indissociable du concept : les concepts n'ont de sens que par leurs interconnexions.

2.2.3.2 Relations sémantiques courantes

Hyperonymie & hyponymie (généralisation & spécialisation)

> *« Il faut en définissant poser l'objet dans son genre et alors seulement y rattacher les différences » (Aristote, Topiques VI, 1)*

L'**hyperonymie** traduit cette dichotomie proposée par Aristote qui est un fondement de notre raisonnement. Cette relation traduit la notion de généralisation du sens. Elle est fréquemment désignée par le nom *« est un » (en anglais « is-a »*). Elle est la base de l'approche aristotélicienne de la définition en genre et différence fortement employée en terminologie (*« genus-differentia »)*. Cette relation est la plus étudiée [Marshman, Morgan et al. 2002], et structure la plupart des ressources lexicales, terminologiques et ontologiques. Cette structure est essentiellement appliquée de façon hiérarchique. Martin Hearst mentionne qu'elle convient *« tout particulièrement à l'extraction semi-automatique des connaissances, car en anglais, les patrons qui expriment la relation hyperonymique n'expriment que cette relation »* [Hearst 1992].

D'un point de vue formel, cette relation est associée à l'héritage et au partage de certaines propriétés, et à la notion d'inclusion dans un ensemble. La relation duale est appelée **hyponymie** (spécialisation). Un terme est hyperonyme (respectivement hyponyme) d'un second s'il possède un sens plus général (respectivement plus spécifique).

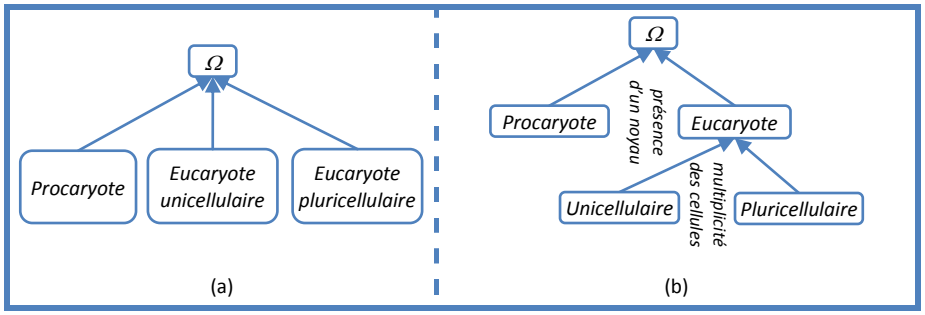


Figure 2.2 – (a) On divise les organismes principalement en trois catégories en fonction de la présence d'un noyau dans la cellule, et de la multiplicité des cellules. (b) Une bonne pratique est de n'utiliser qu'une seule propriété simultanément pour spécialiser un concept en plusieurs hyponymes directs. Ici, le premier niveau spécialise le concept d'organisme suivant le critère de présente d'un noyau, le second niveau est obtenu en séparant les organismes nucléés en fonction de la multiplicité des cellules. Ainsi, la propriété commune de posséder un noyau est héritée dans le (b).

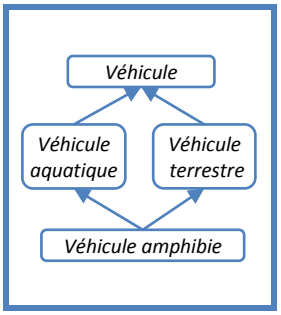


Figure 2.3 – Héritage multiple avec l'hyperonymie.

L'hyperonymie et l'hyponymie sont transitives et antisymétriques. Une bonne pratique durant la structuration par l'hyperonymie est de spécialiser des hyponymes directs par rapport à un seul trait sémantique. Par exemple, la solution (b) est préférable dans la figure 2.2.

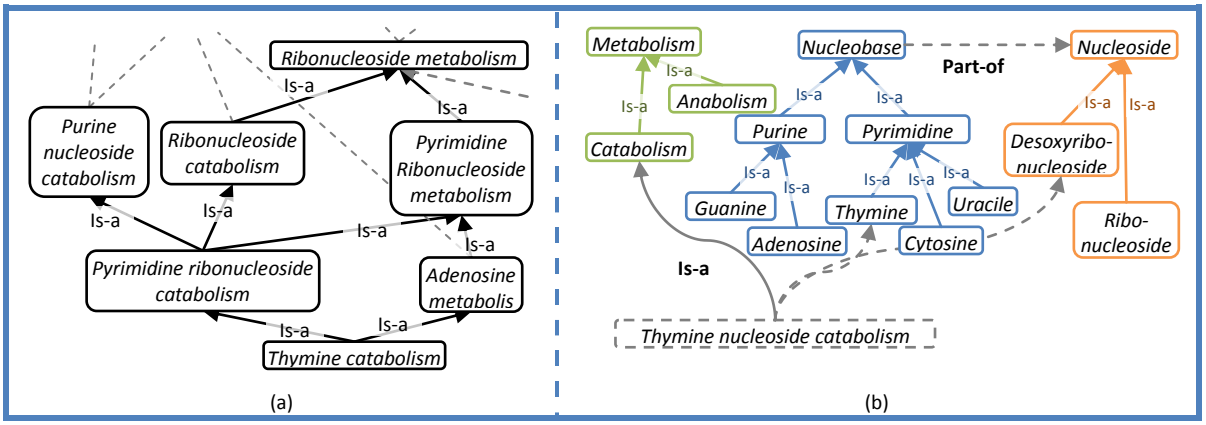


Figure 2.4 – La sous-figure (a) montre la représentation en « DAG[1] » dans la Gene Ontology (cf. annexe C.2 pour un exemple complet issu du portail Amigo). Cette représentation complexe multiplie les concepts possibles liés au métabolisme et aux nucléotides. Dans la sous figure (b) nous proposons une alternative de modélisation décomposant le concept de « thymine catabolism » en trois composantes : le type de nucléotide (ribo ou desoxyribo), la base azotée, et le métabolisme (synthèse ou dégradation). On obtient ainsi des hiérarchies qui permettent la composition de 16 concepts composites. Remarquons que dans la première figure, tous les concepts sont des concepts spécialisant «metabolic process ». Le lien d'héritage entre tous ces concepts se justifie donc. Dans le second, le terme composé n'est pas une spécialisation des trois hiérarchies. C'est une spécialisation du processus catabolique, et une association avec les deux autres concepts. La base azotée peut par ailleurs être considérée comme une partie d'un nucléotide. Les mécanismes à mettre en œuvre pour exploiter cette représentation sont alors bien plus complexes et nécessitent de mettre en place de nouvelles relations biologiques.

La structuration hiérarchique de l'hyperonymie n'est pas sans poser de problème. Un exemple courant illustre l'intérêt de l'héritage multiple (figure 2.3) : le véhicule aquatique et le

[1] Le DAG (directed acyclic graph) est un graphe orienté sans circuit (mais potentiellement avec cycle).

véhicule terrestre sont deux spécialisations de véhicule, alors comment classer le véhicule amphibie ? Cela implique de posséder une modélisation plus fine autorisant la composition de concepts ou une modélisation non strictement hiérarchique (à héritage multiple). Ce second choix a été effectué par le *Gene Ontology Consortium* comme le montre l'exemple ci-dessous concernant le terme *« thymine catabolism »* (figure 2.4). La solution pour obtenir une hiérarchie consiste alors à décomposer un concept considéré comme non atomique. Ceci nécessite la mise en œuvre de mécanismes plus complexes qu'une simple hiérarchie d'héritage. Un exemple complet issu de Gene Ontology est présenté dans l'annexe C.2 (page 284).

Synonymie : équivalence et substituabilité

La **synonymie** est la relation d'équivalence qui associe deux éléments de même sens et de formes différentes. Pour certains, l'identité parfaite des sens n'existe pas dans certaines langues (dont le français). La synonymie est une substituabilité qui dépend d'un contexte [Miller and Charles 1991]. Vincent Nyckees la définit comme *« une relation d'équivalence dont le critère de discrimination est la substitution en contexte » [Nyckees 1998; Schwab 2005].* Violaine Prince la nomme *synonymie relative*, considérant la synonymie parfaite comme *absolue [Prince 1991]*. Didier Schwab précise qu'elle est dépendante d'un contexte [Schwab 2005] et liée à des traits sémantiques [Pottier 1964]. **L'antonymie** de façon générale distingue deux termes dont on peut opposer certains traits sémantiques suivant un axe. Par exemple, on oppose *« terre »* à *« lune »* ou à *« ciel »* en fonction du contexte [Schwab, Lafourcade et al. 2002].

Méronymie : relation entre la partie et le tout

La **méronymie** est la relation entre la partie et le tout. La relation réciproque de la méronymie est appelée **holonymie** : *« noyau »* est méronyme de *« cellule »*, et réciproquement, *« cellule »* est holonyme de *« noyau »*. Moins fréquente que l'hyperonymie dans les terminologies, la méronymie n'en a pas moins été l'objet de nombreux travaux. Achille Varzi a poursuivi les travaux de [Leśniewski 1927; Leonard and Goodman 1940] et établi un cadre formel complet. Winston et al. ont dégagé 7 variantes distinctes de la méronymie en se basant sur quatre propriétés de cette relation (cf. table 2.5) [Winston, Chaffin et al. 1987; Chaffin and Herrmann 1988]:

- *fonctionnalité* (la partie remplit un rôle fonctionnel dans le tout),
- *séparabilité* de la partie du tout,
- *homéomérie* (ou similarité) : *« les parties dites homéomères sont matériellement identiques entre elles et à leur tout : un morceau de tarte est encore de la tarte, un grain de sel est encore du sel, etc. »* [Van Campenhoudt 1994]. Une relation homéomère est réflexive.
- *Simultanéité* : tous les composants sont décrits en même temps. Par exemple, il n'y a pas de simultanéité entre les étapes qui constituent une course, alors qu'il y a simultanéité entre les roues et le volant de la voiture.

Intitulé	Exemple	Fonctionnalité	Homéomère	Séparabilité	Simultané
Composant d'un objet	Roue de la voiture	+	-	+	+
Élément d'un ensemble	arbre de la forêt	-	-	+	+
Portion d'une masse	grain de sel	-	+	+	+
Matière d'un objet	Bois d'une porte	-	-	-	+
La localisation	Banc dans la mer	-	+	-	+
Caractéristique d'une activité	Changer de direction dans un slalom	+	-	+	-
La phase d'une activité	Le weekend dans la semaine	+	-	-	-

Table 2.5 – Ce tableau récapitule les relations de méronymie selon [Winston, Chaffin et al. 1987] & [Chaffin and Herrmann 1988], extrait de [Van Campenhoudt 1994].

Une autre approche rencontrée est celle d'UML [Booch, Jacobson et al. 2004] : deux relations permettent de représenter la méronymie : l'agrégation et la composition. La principale distinction relève de la durée de vie de l'objet en UML. Cette notion est relative à des objets, pas systématiquement à une sémantique de la « partie ». Par exemple, le noyau, la membrane et le cytoplasme sont des parties de la cellule. Lorsque la cellule meurt, son noyau, sa membrane ou son cytoplasme n'a plus d'existence propre. Leurs durées de vies sont liées. Cette notion est proche de la séparabilité de [Winston, Chaffin et al. 1987] et [Chaffin and Herrmann 1988].

La méronymie est souvent considérée comme hiérarchique. Il existe cependant des cas plus problématiques à modéliser : par exemple, une cellule peut être considérée comme faisant partie d'un organe et d'un tissu, mais le tissu n'est pas une partie d'un organe, et la réciproque est toute aussi fausse (cf. figure 2.6). Il est nécessaire de procéder à des raffinements soit en associant le tissu à une notion de type ou de généralisation, soit en proposant plusieurs relations distinctes « partie-tout ». Olivier Dameron rend compte, dans sa thèse, de nombreux problèmes de ce type en anatomie [Dameron 2003]. Cette démarche est adoptée par Sager [Sager 1990] dans le domaine industriel et chimique, par Varzy concernant le problème de la localisation (méréotopologie)[Varzi 1996; Varzi 1996] ou encore par Le Ber en sciences de l'information géographiques [Le Ber and Napoli 2005].

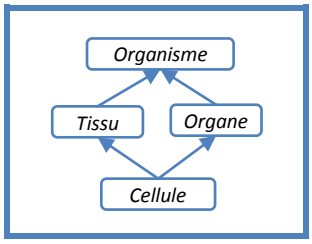


Figure 2.6 – Exemple de hiérarchie non respectée dans la relation « partie-de ».

La question *« Tel objet est-il une partie de tel autre ? »* semble triviale pour certains. Pourtant sa représentation est complexe et source de confusion avec l'hyperonymie. D'une part, l'homéomérie (un grain de sel est du sel), induit un héritage de certaines propriétés. D'autre part, les deux relations sont associées à la notion d'inclusion. Dans le cas de la méronymie, il s'agit d'une inclusion dans un tout ou un ensemble, dans le cas de l'hyperonymie, il s'agit de l'inclusion du point de vue d'un ensemble de propriétés. Cette distinction n'est pas simple à mettre œuvre.

2.2.3.3 Raffinements : domaines & usages.

Les relations précédentes sont les plus courantes. La synonymie est généralement rattachée à des problèmes lexicaux. L'hyperonymie et la méronymie sont les plus utilisées. Il existe de nombreuses autres relations issues de raffinements. Sager s'intéresse à des domaines industriels [Sager 1990], Dameron à l'anatomie [Dameron 2003], Bodson au domaine pharmaceutique [Bodson 2004], Natalya Fridman Noy à la modélisation qualitative des processus, (la *« physique naïve »* ou *« qualitative »*)[Noy 1997], Allen aux relations spatiales et temporelles [Allen 1984], etc. [Gómez, Moreno et al. 2000] contribuent en approfondissant le mécanisme de conceptualisation. Ils placent la relation sémantique comme indissociable du concept : le concept n'a de sens que par ses interconnexions. Ils proposent de classer les relations qui existent entre des concepts suivant 11 types (cf. table 2.7). Ces relations sont cependant redondantes et non réellement structurées. Noy propose au contraire une taxonomie représentées dans la figure 2.8 [Noy 1997]. UMLS qui intègre plus d'une centaine d'ontologies (cf. 57) contient 58 relations sémantiques différentes, non redondantes représentant des relations sémantiques mais aussi des relations spécifiques du domaine : diagnostique, symptôme, composition chimique, effet, etc. [Bodenreider 2004]. Le contenu d'UMLS est détaillé en annexe C.2.2 (page 289).

Type de relation	Exemple	Type de relation	Exemple
Causale	A nécessite B A cause B	**Fonctionnelle**	A permet B A nécessite B
Taxonomique	A est un B A peut être classé comme B, C ou D	**Chronologique**	A avant B A pendant B Relations d'Allen [Allen 1984]
Structurelle	A partie de B A disjoint de B	**Similarité**	Valve A is open = admit petrol
Topologique	A à droite de B A sur B, A dans B , A contre B	**Conditionnelle**	Pour des particules de faible masse fortement accélérées, appliquer les hypothèses quantiques
Dépendance	A partie de B A responsable de B	**Objectif**	Les codes ont été inventés par les humains pour transmettre les idées
Equivalence	bénéfice = recette - dépenses		

Table 2.7 – Les types de relations sémantiques selon [Gómez, Moreno et al. 2000].

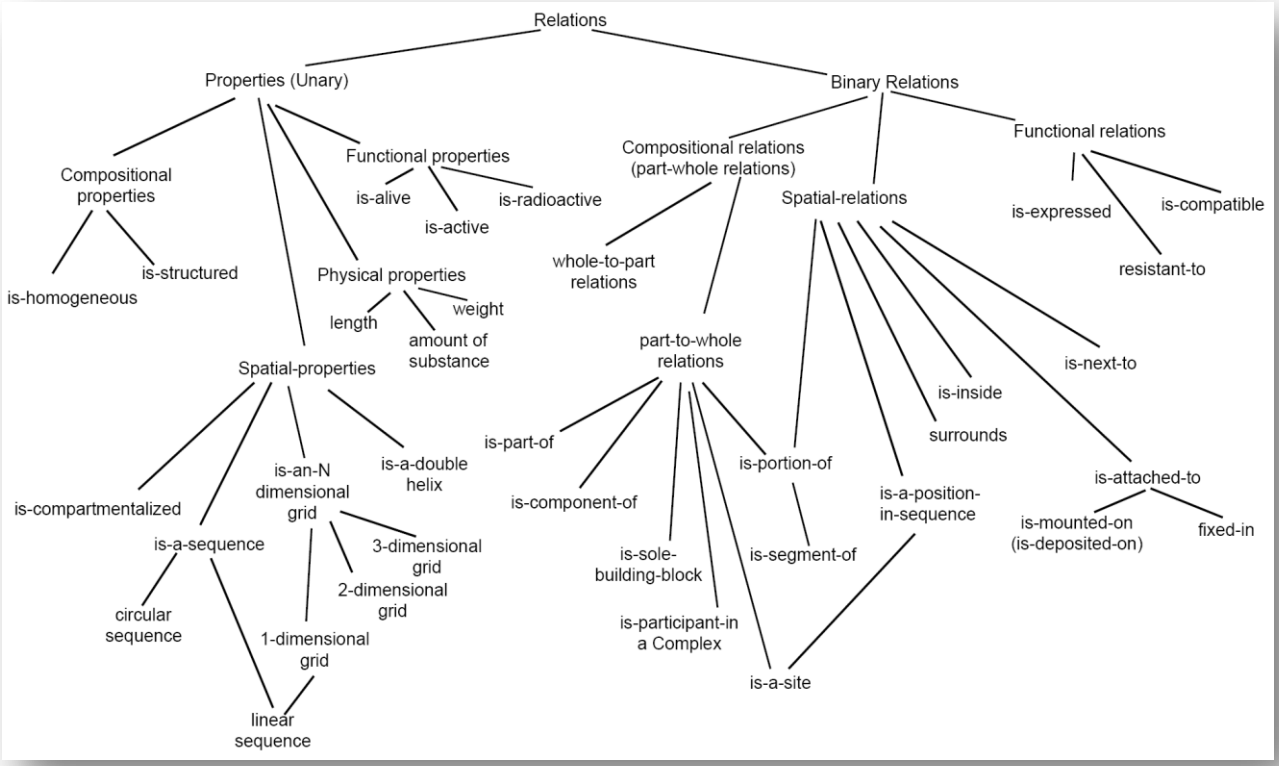


Figure 2.8 –Taxonomie des relations de [Noy 1997].

2.2.4 Types de ressources pour la modélisation

Les ressources lexicales contiennent les mots et/ou lemmes d'une langue. Par exemple, des dictionnaires rassemblent des lemmes et leurs formes fléchies. Ils sont employés comme aide à l'apprentissage de l'orthographe [Arrivé 2006], comme ressource pour la correction ou la vérification automatique de l'orthographe dans des éditeurs de textes [ABU], ou encore pour l'analyse automatique et morphosyntaxique de textes [Chauché 1990]. Le **lexique** ajoute généralement au dictionnaire des définitions. Les formes les plus fréquentes de dictionnaires sont les dictionnaires monolingues et multilingues de la langue. Ce sont donc des ressources langagières. Ils définissent la signification des mots, leur prononciation, leur étymologie, leur

orthographe et leur usage. Pour plus d'informations sur la *« dictionnairique »*, nous renvoyons le lecteur à [Mangeot-Lerebours 2001]

Les ressources terminologiques ajoutent la volonté de normaliser le vocabulaire d'un domaine. Le dictionnaire a pour rôle de définir la signification d'un mot et limiter les ambiguïtés. La terminologie identifie les concepts avec un terme unique et s'abstrait du problème de la synonymie. C'est la fonction de contrôle du vocabulaire. Rastier et Zweigenbaum par exemple conçoivent la terminologie comme permettant une *« médiation entre la pensée rationnelle et le langage » [Rastier 1995]*. *« Le terminologue normalise l'expression des concepts du domaine en fixant les termes qui le désignent »* [Zweigenbaum 1999].

Nous venons d'introduire la notion de vocabulaire contrôlé. Des ressources terminologiques (et ontologiques) sont ainsi souvent désignées par leurs objectifs ou leurs usages dans un problème précis. Cette désignation varie en fonction de la communauté. On emploie ainsi les termes *index*, *thésaurus*, *vocabulaires contrôlés*, *taxonomies*, *lexiques*, *classifications*, *taxonomies*, etc. Ces termes ne s'opposent pas *a priori.*

- l'**index** trouve son origine dans la grande bibliothèque d'Alexandrie. Il permettait une recherche par auteur ou par matière. L'index est constitué d'un ensemble de termes auquel est associée une table de correspondance avec les occurrences (ou instances) de ces termes. L'index est une donnée redondante et alternative qui accélère l'accès à une information (dans un corpus ou dans un document) grâce à un critère de tri. Le plus souvent, il n'interdit pas la présence de synonymes. A la fin d'un livre ou sur un portail documentaire, l'élément est le mot ou le terme. Dans une base de données, le principe est le même, il permet d'accéder à des tuples à partir d'un attribut.
- le **vocabulaire contrôlé**, comme nous l'avons indiqué précédemment, veut normaliser la terminologie d'une communauté. D'une part, il vise à supprimer les problèmes liés à la synonymie et impose l'emploi d'un signifiant unique pour tout signifié. D'autre part, ce vocabulaire doit être manipulable par tous les membres de la communauté : sa taille est donc limitée afin de répondre au besoin de chacun tout en évitant que chacun ait ses propres termes. Plus formellement, il s'agit d'établir une bijection entre l'ensemble des concepts et l'ensemble des termes qui les désignent. Ces ressources permettent de palier les problèmes liés à la synonymie et à la polysémie (ou l'homonymie)[1]. L'intention est le plus souvent de faciliter le traitement automatique d'information, ou la communication entre plusieurs personnes et/ou systèmes d'informations (interopérabilité).
- la **classification**, terme généralement employé par des documentalistes, est une hiérarchie de termes visant à classer les documents en catégories, avec une approche dichotomique [Mcilwaine 2000; Béthery 2005]. Cette classification repose souvent sur l'hyperonymie, mais la structure diffère parfois, et peut même s'avérer hétérogène.
- le **thésaurus**, est une ressource thématique contenant une hiérarchie de descripteurs, parfois complétée de relations transversales. Les termes y sont indexés, reliés à d'autres termes qui leur sont thématiquement relatifs. Sa structuration n'est pas forcément homogène, et ne repose pas forcément sur l'hyperonymie mais sur un découpage thématique. Ce type de ressources est, lui aussi, généralement associé à des problématiques de recherche d'information (indexation), et a été employé notamment dans le cadre de la représentation sémantique lexicale [Lafourcade, Prince et al. 2002]. Alors que la classification permet l'organisation des cotes et rayonnages du centre de documentation, le thésaurus est un outil facilitant la recherche et l'indexation.
- la **taxonomie**[2] (ou taxinomie), est une hiérarchie employée pour la classification ou la structuration dans un contexte plus large que celui de la documentation. Alors que la classification est une approche pragmatique visant à classer des éléments et à naviguer au

[1] La polysémie et l'homonymie sont deux phénomènes comparables : un terme possède plusieurs significations. La polysémie résulte de son étymologie, l'homonyme est le résultat d'une convergence d'écriture aléatoire.

[2] du grec ταξινομία : *taxis* (rangement) et *nomos* (loi) [Wiktionnaire].

sein de leur ensemble, la taxonomie se veut plus modélisatrice. Elle est un outil conceptuel permettant d'expliquer ou synthétiser un domaine, une organisation, et d'aider à la décision.

L'ontologie est une représentation conceptuelle par rapport à la terminologie. Les fonctions précédentes peuvent être relatives à une terminologie ou une ontologie. Les ressources ontologiques se distinguent des ressources terminologiques par leur volonté de représenter les concepts profonds d'un domaine, indépendamment des termes qui peuvent leur être associés [Mizoguchi 2004] ; Bruno Bachimont reprend Guarino et parle d'*engagement ontologique* [Guarino 1994; Bachimont 2000]. Parmi les multiples définitions existantes, celle de Gruber fait référence [Gruber 1993] : « *Une ontologie est la spécification explicite d'une conceptualisation. […] Une conceptualisation est une vue abstraite et simplifiée du monde que nous souhaitons représenter suivant certains objectifs* ». R. Mizoguchi la présente ainsi comme le niveau *meta* d'une base de connaissances. L'engagement conceptuel lui permet de s'affranchir de la barrière linguistique, ce qui évoque de façon immédiate un intérêt pour le multilinguisme [Corby, Dieng-Kuntz et al. 2006]. Cependant, elle n'en est que partiellement affranchie : les concepts sont *« le produit de l'instauration des termes »* [Rastier 1995]. La notion d'engagement conceptuel n'est ni binaire ni simple à évaluer. Certaines ontologies ou terminologies ont des usages, des structurations et des formalisations comparables. R. Mizoguchi nomme ainsi les terminologies des « ontologies de surface » [Mizoguchi 2004]. L'ontologie hérite du passé de la communauté Intelligence Artificielle et des logiques de description notamment. Elle est donc souvent définie comme formelle, permettant des raisonnements logiques [Baader, Calvanese et al. 2003], probabilistes [Ding and Peng 2004], quantitatifs [Fall, Marland et al. 2002], spatiaux et temporels [Stock 1998], etc. Les travaux sur les langages d'implémentation d'ontologies s'attachent simultanément à l'expressivité et à la calculabilité du formalisme [McGuinness and van Harmelen 2004], et certains environnements d'ingénierie d'ontologies proposent des fonctionnalités de vérification de consistance de l'ontologie [Haarslev and Möller 2001; Sirin, Parsia et al. 2006; Tsarkov and Horrocks 2006].

Alors que la terminologie reste à un niveau informationnel, l'ontologie, de par son engagement conceptuel, modélise le monde en accord avec la représentation mentale qu'en a l'homme. [Gómez, Moreno et al. 2000] définit cette conceptualisation comme un travail d'analyse et de synthèse : l'analyse identifie les entités du domaine, la synthèse les structure. Définie comme une vue profonde du monde, elle est systématiquement associée à un contexte : un domaine et une application de façon générale, dans le contexte biomédical à un organisme, un dispositif expérimental, etc.

2.2.5 Quelques exemples de ressources

NCBI Taxonomy

NCBI Taxonomy représente un arbre de classification des espèces vivantes [Wheeler, Barrett et al. 2006]. Cette taxonomie est interrogeable au travers du portail Entrez. Son intégration au sein de nombreux portails et sa taille font qu'elle est souvent considérée comme une base de données et non une ontologie.

Gene Ontology – GO

Nous avons déjà mentionné l'exemple de Gene Ontology [Ashburner, Ball et al. 2000; Consortium 2001; Consortium 2006]. Cette ontologie représente maintenant un standard concernant l'annotation des gènes et de leurs produits. Elle est indépendante d'un organisme, mais contient certains termes spécifiques à certains organismes (plantes, bactéries, etc.). Sa conception est le plus souvent basée sur l'insertion d'ontologies existantes. Le contenu est rigoureusement vérifié par expert et corrigé si nécessaire.

GO se décompose en trois ontologies qui modélisent les processus biologiques (13 533 termes), les fonctions moléculaires (7609 termes) et enfin les composants cellulaires

(localisation, 1966 termes). Cela représente un total de 23 128 termes dont 97% possèdent une définition. GO est structurée par deux relations sémantiques, *« est un »* (majoritairement) et *« part-of »*. Une évolution appelée GONG (GO Next Generation) se focalise sur des sous ensembles et migre dans une logique de description (OWL) afin de pourvoir enrichir automatiquement les relations [Wroe, Stevens et al. 2003].

Amigo est le portail principal qui permet d'accéder à GO. Il contient notamment de nombreuses références croisées vers d'autres systèmes d'information. GOA (Gene Ontology Annotation) est un portail dédié à l'annotation utilisant Gene Ontology. Enfin, le téléchargement de Gene Ontology est proposé avec de nombreuses références croisées correspondant à ces mêmes annotations. Ces annotations sont disponibles en téléchargement sur le portail de GO. Cette ontologie est omniprésente dans le quotidien du chercheur en biologie moléculaire et cellulaire.

MeSH

Le MeSH (Medical Subject Headings) est le vocabulaire contrôlé conçu par la bibliothèque de médecine des Etats-Unis [Schulman 1997; Kostoff, Block et al. 2004]. Il est conçu pour l'indexation des documents dans PubMed. L'utilisation est avant tout celle d'un vocabulaire contrôlé et cette ressource est structurée par une relation de spécialisation. On y distingue les descripteurs primaires, au nombre de 23 000 et les descripteurs secondaires au nombre de 150 000. Plus de 100 000 termes supplémentaires sont rattachés par relation de synonymie. Le MeSH est traduit en plusieurs langues, notamment le français, une tâche réalisée par l'INSERM.

SnomedCT – the Systematized Nomenclature of Medicine – Clinical terms

Plus spécifique au domaine médical, SnomedCT est utilisé pour l'indexation de rapports d'actes médicaux. Il contient près de 400 000 concepts et constitue, lui aussi, une référence dans son domaine.

2.3 Intégration de données

L'intégration de données est un domaine relatif aux bases de données et à leur fouille qui est apparu au début des années 90. On peut considérer deux domaines de recherche qui la concernent : *l'interopérabilité des systèmes* qui vise à définir des normes, standards, protocoles afin de faciliter la communication entre les systèmes, et *l'intégration* à proprement parler qui vise à unifier et centraliser l'accès à plusieurs ressources. Dans cette section, nous présentons dans un premier temps les différentes sortes d'hétérogénéité qui sont problématiques. La mise en œuvre de standards, de normes, de protocoles contribue à l'interopérabilité des systèmes et permet de résoudre une partie du problème. Cependant, une hétérogénéité dans les schémas des systèmes d'information et dans les données reste présente et nécessite la mise en œuvre de systèmes d'intégration. Deux approches dans la conception d'un système d'intégration existent : l'approche médiateur (virtuelle) et l'approche entrepôt (matérialisée). La dernière partie de cette section détaille une autre approche de l'intégration qui poursuit l'objectif d'être orientée vers l'utilisateur final dans le cadre d'un support à l'analyse de ses données.

Comme nous l'avons fait concernant les domaines précédents, cette section définit un ensemble de termes et présente les différentes approches existant dans la communauté. Les principaux systèmes liés à la bioinformatique son cités en parallèle.

2.3.1 Généralités

L'intégration de données concerne essentiellement deux problématiques : l'accès unifié à des ressources distantes (distribuées), et à des données hétérogènes. La première a conduit à

l'apparition d'un grand nombre d'outils logiciels permettant un accès à des ressources distribuées. L'hétérogénéité reste un problème sur lequel se concentrent de nombreux efforts.

Dans cette section, nous présentons dans un premier temps la terminologie générale du domaine : dépôt, système d'intégration, interopérabilité, etc. Nous présentons ensuite les différents niveaux d'hétérogénéité qui posent problème et les solutions adoptées, d'une part en termes de standardisation, d'autre part dans la conception de systèmes d'intégration.

2.3.1.1 Distribution, Complémentarité et hétérogénéité

Distribution et complémentarité : Sources primaires et secondaires

Nous l'avons vu, le domaine biomédical possède un nombre important de bases de données et en continuelle augmentation (cf. figure 1.24 page 33). Dans sa thèse, Sarah Cohen Boulakia propose de distinguer les sources primaires des sources secondaires [Cohen-Boulakia 2005].

Les sources primaires (ou dépôts de données) contiennent des données brutes : ces données ne sont pas soumise à un processus de validation. Par exemple, GenBank/EMBL/DDBJ sont des dépôts de séquences génomiques, GEO/ArrayExpress stockent des données d'expressions issues de puces à ADN. Ces sources servent en particulier de référence pour la communauté scientifique. Avant de publier les résultats d'un séquençage ou d'une analyse de données d'expression, le chercheur doit partager ses données dans l'une de ces ressources.

Les sources secondaires sont construites à partir des données des sources primaires. Les données sont dans un premier temps filtrées et contextualisées (vis-à-vis d'un domaine, d'une maladie, d'un organisme, etc.). Dans un second temps, elles peuvent être nettoyées et épurées (*« curated »*) suivant une perspective de qualité des données. Ce nettoyage peut être issu d'une procédure automatique ou manuelle. Par exemple, RefSeq contient les données de GenBank auxquelles on a ôté les séquences redondantes [Pruitt, Tatusova et al. 2005]. De même, UniParc contient les séquences non redondantes d'UniProt. Une autre procédure automatique consiste à créer des regroupements de séquences similaires : c'est le cas d'UniGene créée à partir des séquences contenues dans GenBank et d'UniRef pour les séquences d'UniProt. UniRef propose trois sources : UniRef100, UniRef90 et UniRef50. L'indice numérique correspond au pourcentage d'identité entre les séquences d'un même groupe. D'autres portails comme PlasmoDB proposent un niveau de précision plus élevé avec un nettoyage supervisé par un expert (manuel) [The Plasmodium Genome Database Collaborative 2001]. Le critère de filtre n'est alors plus uniquement la similarité entre deux séquences.

Les dépôts de données sont initialement faiblement structurés. Ce sont des banques de données dans lesquelles chaque séquence est stockée sous forme d'un fichier. Alors que les sources primaires sont parfois des banques de données, les sources secondaires sont généralement structurées autour de schémas dans de véritables bases de données.

Hétérogénéité des données

Ces sources de données sont particulièrement nombreuses, réparties et hétérogènes. Leur nombre est tel qu'il est aujourd'hui critique pour le biologiste d'avoir connaissance des sources qui lui sont utiles. Nous avons déjà abordé le problème structurel : certaines sources sont des banques de données, d'autres sont des bases des données. Au sein même des bases de données, les formats, schémas et modèles et données diffèrent. On distingue plusieurs niveaux d'hétérogénéité qui posent chacun des problèmes particuliers [Wiederhold 1992; Davidson, Overton et al. 1995; Baril 2003; Cohen-Boulakia 2005].

Les conflits structurels sont opposés aux conflits sémantiques. Les conflits structurels sont relatifs aux choix d'architecture, de format, de protocole, de serveur de base de données, de paradigme de modélisation, de langage de requête, etc. Pour résoudre ces conflits, des outils existent et permettent une bonne interopérabilité : des intergiciels permettent de s'abstraire du réseau, du langage, du SGBD, du protocole, etc. Les paradigmes de modélisation employés dans les sources peuvent différer (hiérarchique, relationnel, objet, logique de description, etc.). Pour

préserver l'information, il est nécessaire d'employer un paradigme ayant un pouvoir d'expression au moins équivalent à ceux des paradigmes des sources. Les paradigmes les plus fréquents sont le relationnel et l'objet. Il existe des algorithmes de traduction de schéma dans les deux sens.

L'hétérogénéité sémantique se manifeste à deux niveaux : des conflits entre les schémas des sources, ou des conflits entre leurs données (tuples ou instances). Les entités et attributs d'un schéma peuvent être absents dans l'autre. Ce qui est un attribut dans l'un peut être une entité dans l'autre. Les associations et contraintes entre entités varient, tout comme les types et domaines des attributs. Dans le paradigme objet, les liens d'héritage introduisent des variations supplémentaires. Toutes ces différences peuvent s'avérer des contradictions et résultent essentiellement de points de vues : veut-on représenter le locus ? Les données initiales sont elles structurées ? Etc. Il s'agit aussi de choix techniques qui influent sur le schéma. Dans la 2.3.1.2 (page 5755) nous détaillons UMLS qui, pour des raisons d'optimisation de temps de recherche, regroupe quatre entités dans une même table. En cela, UMLS ne respecte pas la deuxième forme normale de Codd (cf. annexe A.1.2 page 260).

Comme nous l'avons introduit au début de ce chapitre, la modélisation durant la conception d'une ontologie ou d'un schéma de base de données est une tâche dépendante d'un contexte, d'un domaine et d'une application. Il est inévitable d'être confronté à une hétérogénéité des schémas lorsque l'on intègre des sources construites suivant différents besoins. Des approches proposent des schémas répondant à de multiples besoins : GUS est réutilisable dans différents contextes de la génomique [Davidson, Crabtree et al. 2001]. La contrepartie est la complexité du schéma ; GUS propose plus de 300 tables.

Enfin, en bioinformatique, il est courant d'avoir des données (instances) contradictoires ou redondantes. Cela provient dans un premier temps de la réalité expérimentale : Il arrive parfois que des équipes travaillent sur des organismes communs et proposent simultanément de nouvelles séquences. De plus, les dispositifs expérimentaux ne sont pas parfaitement fiables et les expériences ne sont pas parfaitement répétables. Différentes expériences peuvent concerner des souches différentes qui ne sont pas spécifiées dans le formulaire. La manipulation du formulaire peut également donner lieu à des erreurs : un clic de trop peut engendrer la soumission multiple d'une même séquence. Pour toutes ces raisons, on trouve ainsi des séquences identiques (synonymes) et des séquences différentes portant un même nom (homonymes). Les annotations sont parfois insérées manuellement après une expertise, parfois produites automatiquement par des outils prédictifs. La fiabilité relative des différentes origines des annotations les rend complémentaires. Le système doit respecter une traçabilité de l'origine de l'information afin que l'expert puisse décider.

2.3.1.2 Interopérabilité et standardisation

Interopérabilité syntaxique

La figure 2.9 résume les différents niveaux d'hétérogénéité et propose quelques exemples. A ces problèmes d'hétérogénéité correspondent différentes problématiques informatiques et différents outils apportant des réponses partielles. Les problèmes structurels évoqués relèvent majoritairement du problème de l'interopérabilité, définie comme suit :

> *« L'interopérabilité est le fait que plusieurs systèmes, qu'ils soient identiques ou radicalement différents, puissent communiquer sans ambiguïté et opérer ensemble. » (Wikipédia)*

Hétérogénéité structurelle		Accès	*CGI, HTTP, FTP, ODBC, JDBC, RMI, CORBA, etc.*
		Syntaxe/format	*XML, ASN.1, HTML, texte brut, GenBank, Fasta, etc.*
		Modèle *(paradigme)*	*Relationnel, objet, relationnel-objet.*
		Structure	*Présence d'un schéma, forme normale, etc.*
		Langage de requête	*SQL, OQL, SparQL, syntaxes propriétaires, etc.*
		Outils et services	*Blast, alignements, visualisation et interactivité, etc.*
Hétérogénéité sémantique	Schémas	Point de vue et niveau de granularité	*Dans SwissProt, le gène n'est qu'un attribut, la protéine est au centre du schéma. Inversement, dans GenBank, c'est la séquence nucléotidique qui est au centre.* *En fonction du schéma, les attributs ne sont pas forcément les mêmes. Dans une BD, l'espèce n'est qu'une valeur textuelle, dans d'autres la taxonomie complète est représentée.* *Les entités et attributs varient (absence, nature, etc.)*
		Définition biologique d'une entité	*Certaines entités ont des significations différentes : dans une BD de séquences, un gène est une séquence, dans une autre, c'est uniquement la partie codante, avant ou après épissage, etc.*
	Instances	Clé/Identification	*Des séquences identiques sont présentes de nombreuses fois (synonymie, redondance), des séquences différentes portent le même nom (homonymie)*
		Points de vue ; signification	*Nombre de bases ou paires de bases, notion de séquence « longue » pour un gène, une protéine, une sonde, séquences « relatives »*
		Vocabulaire	*Les termes ne sont pas les mêmes : « adenosine metabolism » et « adenosine metabolic process », les sources utilisent des terminologies distinctes*
		Valeur	*Un même gène est annoté différemment dans plusieurs sources : une annotation est plus « profonde » ou contradictoire, obtenues par des techniques différentes (manuelles et automatiques par exemple), problème des unités, etc.*

Figure 2.9 – Exemples d'hétérogénéité entre les sources d'informations.

L'interopérabilité consiste donc à résoudre certains conflits structurels. On peut alors dissocier plusieurs problématiques (non exhaustives) :

- Les protocoles réseau poursuivent l'objectif d'abstraire le développeur et l'utilisateur des problématiques liées à l'hétérogénéité du réseau et à la gestion de la communication de bas niveau. On retrouve ainsi les standards comme FTP, HTTP, CGI.
- Les formats de codification et d'échange de données sont par exemples : Unicode pour les caractères, XML pour les documents structurés, RDF et OWL (dérivé d'XML) pour les métadonnées et les ontologies.
- Les langages d'interrogation permettent l'interrogation homogène de systèmes : SQL pour les SGBDR [Codd 1970], OQL pour les SGBDO, SparQL pour interroger du RDF dans Jena, etc. Des API permettent cette homogénéité dans le développement.

- Certains intergiciels permettent l'abstraction du réseau à un plus haut niveau dans le développement d'applications. ODBC et JDBC homogénéisent l'accès aux SGBD au travers d'API. C-JDBC se spécialise sur des données réparties ou distribuées [Cecchet 2004]. CORBA, RMI fournissent la possibilité de partager et distribuer des objets. ProActive ajoute un niveau d'abstraction en abordant la problématiques des composants distribués [Baduel, Baude et al. 2006]. JDO et Hibernate proposent un mécanisme de persistance d'objets en Java, Jena propose l'équivalent pour des modèles ontologiques en RDF ou OWL [McBride 2001].

Interopérabilité sémantique

Les outils précédents permettent principalement de gérer l'hétérogénéité structurelle et syntaxique. L'hétérogénéité sémantique considère les différences en termes des données ou de leur modélisation (les schémas). D'après Lincoln Stein, le problème de l'intégration n'est alors pas tant un problème technique qu'un problème sociologique (nous revenons sur cette considération dans la section 3.3.1 page 87)[Stein 2003]. Il explique que l'intégration ne peut se faire que si les concepteurs des bases de données, les fournisseurs des données se positionnent collectivement et coopèrent. La communauté doit au préalable instaurer des standards d'interopérabilité acceptés. Dans son article, il met notamment en avant le problème de la nomenclature des entités biologiques (gènes, protéines, etc.). Nous discutons plus amplement ce problème dans le chapitre qui suit.

On peut ainsi considérer plusieurs phénomènes de standardisation dans cette perspective. A un niveau assez technique, l'ODMG propose une initiative, le LSID (Life Science Identifier)[Martin, Hohman et al. 2005], qui porte le principe de l'URI au domaine biomédical : une indentification universelle des ressources. D'un point de vue plus conceptuel, les ontologies sont aujourd'hui fréquemment employées comme vocabulaire contrôlé. Gene Ontology fait, par exemple, autorité dans l'annotation de gènes, protéines et transcrits en biologie moléculaires. SnomedCT fait référence dans le domaine médical. Ces ressources sont multiples, mais des alignements existent au sein d'entrepôts : UMLS (décrit dans la section suivante) contribue à l'interopérabilité sémantique de systèmes employant des ontologies ou terminologies différentes. Enfin, ces terminologies normalisent le vocabulaire et sa définition, mais n'interviennent pas dans la nomenclature des gènes et protéines par exemple. Cette difficulté ancienne est par exemple détaillée par T.K. Jenssen [Jenssen, Laereid et al. 2001]. Face à ce constat, des initiatives apparaissent comme Hugo, le standard de nomenclature des gènes chez l'homme.

UMLS

Malgré la structuration de la communauté qui tend à imposer des standards d'interopérabilité, les ressources termino-ontologiques sont de plus en plus nombreuses car spécifiques à un domaine et une application. Pour permettre à plusieurs systèmes d'information de s'échanger de l'information avec précision, il est nécessaire que les différents vocabulaires des agents communicants puissent être alignés (mis en correspondance). Les motivations sont alors les mêmes que pour l'intégration de données : ces alignements sont très coûteux. Plutôt qu'une approche point-à-point, UMLS (*Unified Medical Language System*) centralise cet alignement en créant un niveau conceptuel (pivot) et un niveau terminologique. La terminologie qui s'aligne avec le niveau conceptuel est interopérable avec celles précédemment alignées. Ce niveau conceptuel est appelée *MetaThesaurus* pas ses auteurs.

Figure 2.10 – Structure générale d'UMLS avec cardinalités.

Nous considérons UMLS comme un entrepôt de ressources terminologiques et ontologiques. Il comporte actuellement plus d'une centaine d'ontologies. Son schéma général découpe en réalité la médiation en quatre niveaux (schéma dans la figure 2.10 et exemple dans la figure 2.11) :

- les concepts,
- les termes qui se rapportent à ces concepts,
- les écritures de ces termes (*Strings*) qui prennent en compte flexions, abréviations et autres variations courantes,
- les occurrences (*Atoms*) qui correspondent aux instances des écritures dans les sources (ontologies intégrées).

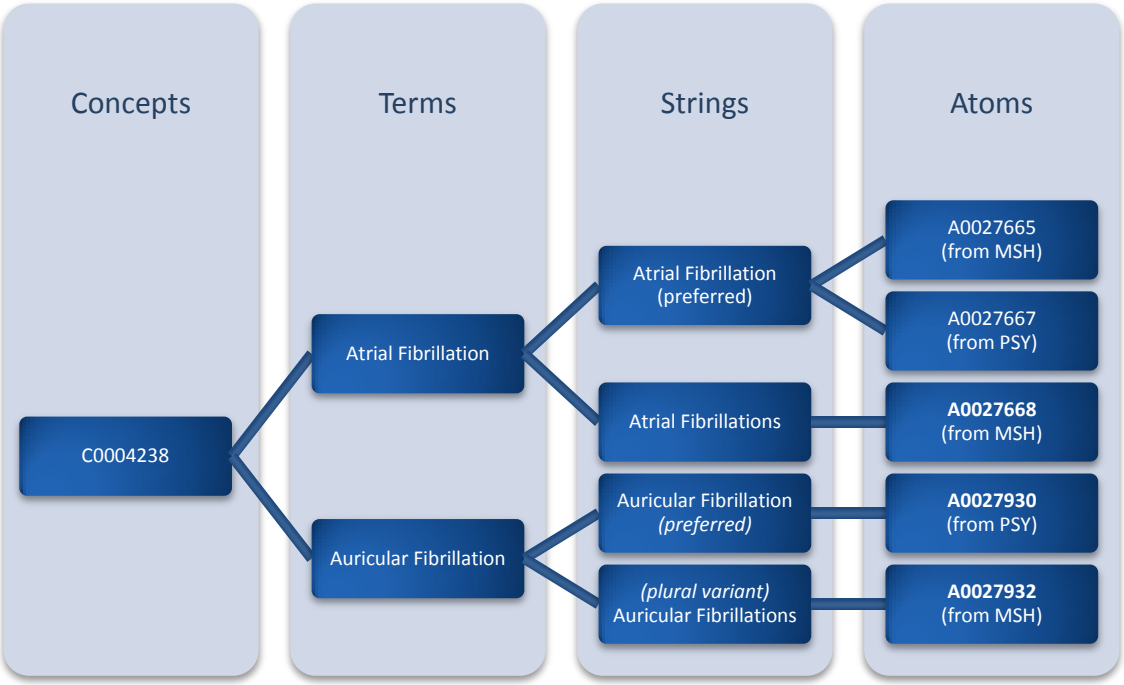


Figure 2.11 – Exemple de contenu d'UMLS (extrait de la documentation officielle).

UMLS propose un outil de navigation et peut être intégré dans un SGBDR (des scripts sont générés pour Oracle et MySQL). Cependant, il faut noter que le schéma relationnel ne distingue pas quatre relations conformément au schéma ci-dessus, mais stocke toutes ces informations dans une relation unique. UMLS ne respecte donc pas la seconde forme normale de Codd, dans le but d'obtenir des performances d'accès optimales. Il contient près d'un million de concepts et près de 5 millions d'atomes. Les données sont mises à jour sous forme de 2 à 4 distributions par an. Lorsque ces distributions sont disponibles, un décalage est généralement déjà présent avec les principales ressources qui évoluent parfois de façon hebdomadaire. Enfin, il est fourni avec un serveur libre du nom d'UMLSKS (UMLS Knowledge Source Server).

Concernant les données contenues, et plus particulièrement leur qualité, de nombreuses anomalies sont présentes. Par exemple certains termes sont des séquences nucléotidiques, etc. Une partie des données provient donc de procédures d'extraction automatique. Il est important de conserver une traçabilité sur les données et leur origine. Certaines ontologies offrent en effet la garantie d'un nettoyage consciencieux par un expert.

UMLS est une ressource qui sépare le niveau conceptuel du niveau terminologique, cependant, les ressources qu'il intègre n'ont pas toujours l'engagement conceptuel souhaité. Dans la pratique, il est préférable de considérer UMLS comme un outil d'interopérabilité et non comme une ontologie (formelle). Une description schématique complémentaire du contenu d'UMLS (sources et relations sémantiques) et des captures d'écran des interfaces utilisateurs (UMLS Browser et UMLSKS) sont présentes dans l'annexe C.2.2 (page 289).

2.3.1.3 Intégration et système d'intégration.

Les standards contribuent à l'interopérabilité syntaxique et sémantique. Il est nécessaire de résoudre les conflits relatifs aux schémas et données. L'intégration de données est généralement définie comme le rassemblement des données provenant de plusieurs sources (distribuées ou hétérogènes). Lorsque le problème relève uniquement de la distribution, et que les données sont structurées et décrites de façon homogène, on parle de « bases de données distribuées ». L'intégration de donnée induit le plus souvent des problématiques d'hétérogénéité. Il existe plusieurs définitions, nous avons retenu celle de Gio Wiederhold :

> *Un système d'intégration de données fournit une vue unifiée de données provenant de sources multiples et hétérogènes. Il permet d'accéder à ces données au travers d'une interface uniforme, sans se soucier de leur structure ou de leur localisation. [Wiederhold 1992]*

Cette définition est à replacer dans le contexte des bases de données. La notion de vue implique la présence d'un schéma (virtuel ou matérialisé) unique. La notion d'accès uniforme signifie la présence d'un langage de requête complet. L'interrogation des données dans ce schéma doit s'effectuer dans une transparence totale vis-à-vis de l'hétérogénéité des données et de leur localisation. Cependant, par rapport à une source locale homogène, une fonctionnalité souhaitable d'un système d'intégration est la traçabilité : on souhaite savoir quelle source a permis d'obtenir la donnée. Dans le cas d'une approche matérialisée, on apprécie également l'historisation des données, de leur mise à jour, de leurs corrections.

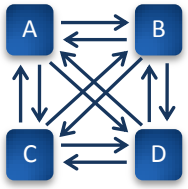


Figure 2.12 – L'intégration point à point. Soient A, B, C et D quatre bases de données, chacune souhaitant être dans la capacité d'interroger toutes les autres. Le nombre de connexions est $n \times (n-1)$ où n est le nombre de systèmes.

Pour intégrer l'information, une approche possible est la connexion point à point (appelée aussi approche fédérée) [Garcia-Molina, Ullman et al. 2002]. Chaque source a une connaissance des autres et établit les connexions nécessaires (figure 2.12). L'ajout d'une $n^{\text{ième}}$ source implique l'implémentation de $(n-1)$ connexions. Cette explosion combinatoire rend cette approche adaptée lorsque l'on a un nombre de systèmes fixe et très restreint.

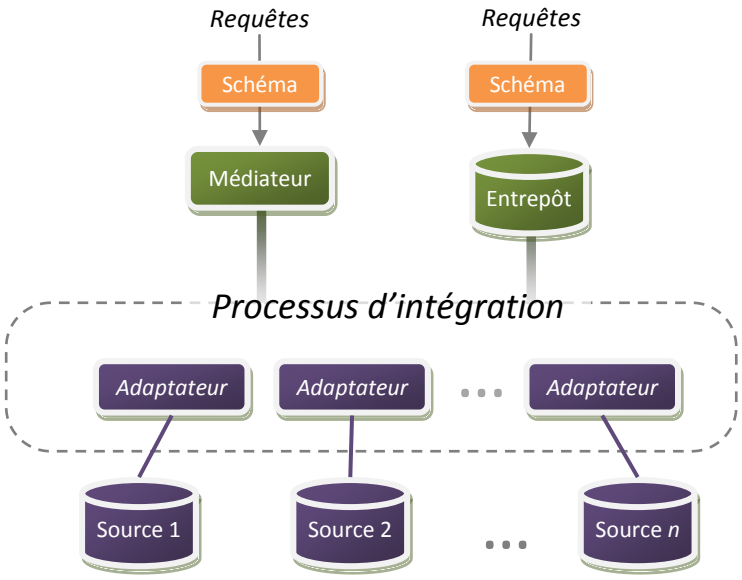


Figure 2.13 – Dans un système d'intégration, l'interrogation se fait par un langage de requête vers un schéma unifié qui permet de s'abstraire de la distribution et de l'hétérogénéité des données. Un adaptateur (« wrapper ») permet l'intégration des données depuis une source. La distinction entre entrepôt et médiateur réside dans la matérialisation ou non des données intégrées.

Pour résoudre ce problème, il est nécessaire d'introduire une centralisation (figure 2.13) : c'est l'intérêt d'un schéma global (aussi appelé schéma médiateur ou intégrateur) qui joue le rôle

de pivot entre les sources dites locales. Wiederhold divise l'intégration en quatre tâches que Susan Davidson précise (Wiederhold Genesereth 1997)[Davidson, Overton et al. 1995] :

- accéder à plusieurs sources et récupérer l'information,
- abstraire et transformer chaque donnée dans un modèle et un schéma commun,
    - traduire les schémas dans un paradigme aussi expressif que ceux des sources locales
    - aligner les schémas au niveau des entités (et des attributs), créer des spécialisations ou généralisations d'entités si nécessaire.
    - Intégrer les schémas à proprement parler, c'est-à-dire créer un schéma global fédérateur (l'approche la plus simple consiste à créer l'union des schémas locaux)
- intégrer les données homogénéisées,
    - transformer les données des schémas locaux vers le schéma global
    - établir la correspondance entre les éléments des différentes réponses.
- réduire les données intégrées par abstraction pour accroître la densité d'information.

La représentation familière d'une entité dans le paradigme relationnel est celle d'une table. Les colonnes sont les attributs, disposés horizontalement sur la première ligne de la table, tandis que chaque instance (tuple, enregistrement) est représentée par une ligne, listée verticalement. Pour ces raisons, l'intégration des schémas est appelée intégration horizontale (ou intentionnelle), et l'intégration au niveau des instances est appelée intégration verticale (ou extensionnelle) [Sujansky 2001; Hernandez and Kambhampati 2004].

Pour classer les systèmes d'intégration, Susan Davidson propose d'utiliser deux axes : le degré de fédération (faible ou fort[1]) et le niveau de matérialisation (*versus* virtualisation) [Davidson, Overton et al. 1995]. Elle précise que ces axes sont méthodologiquement indépendants des systèmes existants que l'on peut citer en exemple. Le niveau de fédération n'influe pas sur le niveau de matérialisation.

Intégration « faible » ou « forte »

Un système est d'autant plus fortement intégré que l'on établit plus de correspondances sémantiques entre le schéma global et les schémas locaux. L'avantage d'un haut niveau de fédération est la présence d'un seul schéma modélisateur et d'un seul langage de requête, répondant à la demande de l'utilisateur, qui est souvent un développeur. Un haut niveau de fédération est cependant coûteux durant toutes les étapes de l'intégration, en particulier durant l'intégration horizontale et verticale. A l'inverse, un système lâche ou faiblement intégré est conçu autour d'un schéma résultant de l'union des schémas locaux. Les avantages et désavantages sont réciproques : dans le système faiblement intégré, l'insertion des données dans le schéma global est immédiate, mais l'exploitation de ces données est plus complexe.

Entrepôt versus médiateur

Le second axe est celui de la matérialisation. Lorsque nous avons présenté les notions élémentaires relatives aux bases de données, nous avons introduit les vues. Les vues sont des sortes de tables virtuelles. Dans le cas d'un système intégrateur, on distingue l'approche entrepôt et l'approche médiateur. L'approche entrepôt stocke (matérialise) les données dans un entrepôt alors que l'approche virtuelle propose des vues non matérialisées sur le schéma global. Jennifer Widom qualifie l'approche médiateur d'*« à la demande »*. En effet, lorsque l'on émet une requête, cette requête est traduite puis propagée vers chaque source. Après réception, les résultats sont fusionnés. L'avantage d'une telle approche est que les données sont systématiquement à jour. Un tel système ne nécessite pas d'investissement important en termes de matériel, de déploiement et d'administration. A l'inverse, J. Widom qualifie l'approche

[1] Respectivement *« tight »* ou *« loose », « lâche »* ou *« serré »*

matérialisée d'intégration *« à l'avance »*. La réactivité et les performances sont supérieures : le calcul lié à l'intégration a été réalisé au préalable et l'existence d'un schéma global matérialisé autorise l'optimisation des requêtes. De plus, cette approche évite tous les problèmes liés à la communication distante et à l'autonomie de la source : les délais de communication distante deviennent négligeables, les performances du système ne sont plus dépendantes des performances et charges des sources, un incident sur le réseau ou dans l'une des sources n'implique pas l'indisponibilité d'une partie du système. Cette approche est donc recommandée dans le cadre d'une fouille de données. Enfin, dans ce type d'approche, la mise en œuvre de jointure et de contraintes entre les sources est moins coûteuse. La limite de l'approche matérialisée réside essentiellement dans le coût d'actualisation des données. Le problème consiste à détecter l'existence d'une mise à jour, détourer le contenu mis à jour puis mettre les données effectivement à jour. Enfin, dans le cas d'utilisation de méthodes de fouille de données de l'entrepôt, il faut ré-exécuter ces procédures, qui idéalement sont incrémentales.

Ces limites correspondent à des difficultés techniques liées à l'intégration matérialisée ou virtuelles. Le problème « sociologique » abordé par Lincoln Stein soulève une question bien plus importante : l'approche virtuelle repose sur l'hypothèse que les systèmes distribués et hétérogènes sont accessibles par un langage de requête complet. Dans la pratique, très peu de systèmes sont accessibles de cette façon. Certains éléments de solution pallient ce problème, par exemple au sein de Discovery Link [Haas, Schwarz et al. 2001].

Autres approches

Les systèmes d'intégration visent à conserver les données avec l'expressivité de leurs sources tout en fournissant un mécanisme de requête unifié et centralisé. Nous pouvons découper ces systèmes en deux catégories disjointes : les entrepôts (approche matérialisée) et les médiateurs (approche non matérialisée). D'autres approches ne fournissent pas les mécanismes de requêtes énumérés précédemment et poursuivent d'autres objectifs. C'est le cas du système à base de liens : il considère une seule entité (le document ou fichier), et des références croisées entre des fichiers. Cette approche est considérée comme lâche. Les données peuvent être matérialisées ou non. Si les documents sont parfois distants, les références croisées sont le plus souvent stockées. La classification des systèmes dans ce mémoire peut donc s'avérer discutable, et plusieurs points de vue coexistent dans la communauté. Celle-ci respecte les classifications proposées par les articles référencés au début de cette section.

De même, les plateformes sont orientées vers l'utilisateur pour fournir des outils adaptés à des problèmes et tâches spécifiques. Elles sont définies en tant que support pour l'analyse : il est possible d'appliquer des procédures diverses, de stocker les résultats intermédiaires, de générer des visualisations, etc. Contrairement aux approches précédentes, si les plateformes intègrent des données hétérogènes, elles se distinguent par l'absence ou l'aspect facultatif de mécanismes de requête. L'ensemble de tous les outils décrits jusqu'ici est structuré dans la taxonomie qui suit (figure 2.14).

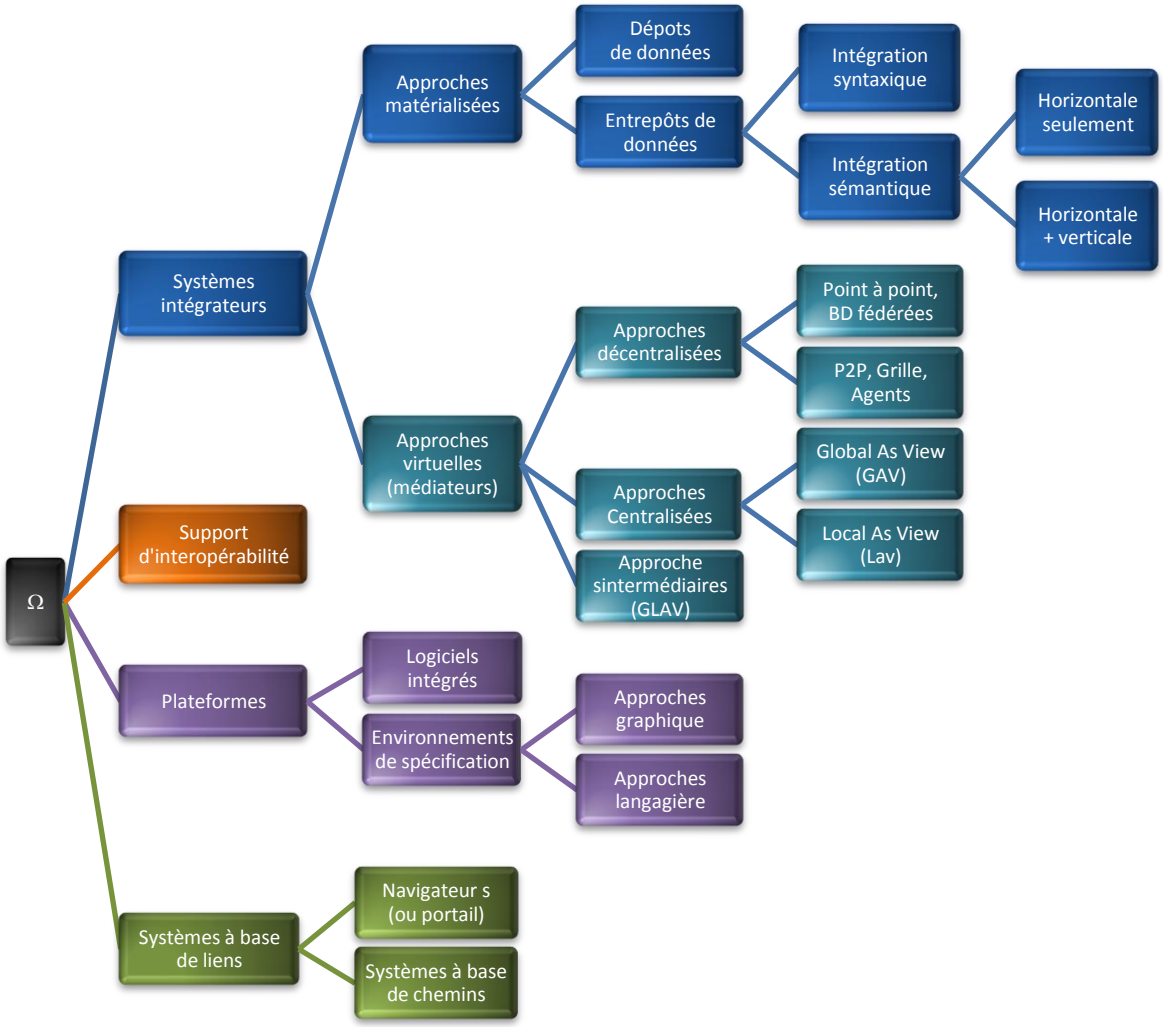


Figure 2.14 –Taxonomie des outils présents dans le domaine de l'intégration de données. On distingue globalement 4 classes :
1. les systèmes d'intégration fournissent une vue unifiée et centralisée sur les données manipulables par un mécanisme de requêtes et respectent l'expressivité des sources.
2. les outils d'interopérabilité (intergiciels, normes et standards, ontologies, etc.)
3. les systèmes à bases de liens relient documents, pages, et diverses entités par des références croisées .
4. les plateformes qui fournissent un support d'analyse spécifique à un nombre restreint de problèmes et de tâches.

2.3.2 Système d'intégration

Un système d'intégration est donc défini comme un système donnant l'impression d'interroger plusieurs sources distribuées et hétérogènes comme un SGBD. Il met donc à disposition de l'utilisateur un langage de requête comme SQL en relationnel ou OQL pour le paradigme objet par exemple. Pour proposer une vue unifiée, il existe deux approches : matérialisée ou virtuelle. La première stocke les données localement. La seconde au contraire ne stocke que quelques métadonnées ou des données temporaires. Chaque requête est traduite et expédiée vers chaque source. Les résultats sont récupérés puis fusionnés. Nous détaillons ces deux approches dans cette section.

2.3.2.1 Approche matérialisée : l'entrepôt

Définition et caractéristiques

L'entrepôt, dans son sens commun, est un bâtiment industriel permettant le stockage et la manipulation de grandes quantités de matière ou produits. L'entrepôt en informatique est, de la même façon, un lieu de stockage centralisé adapté à la manipulation d'une grande quantité de données. Il résulte de différents besoins. Les projets de séquençage de génomes qui sont apparus nécessitaient un stockage structuré de données relatives aux séquences. En parallèle, l'informatique d'entreprise a exprimé le souhait d'outils d'aide à la décision, incarnés aujourd'hui principalement par les « OLAP » (*« On Line Analytical Processing »*). Enfin, un troisième secteur où les entrepôts sont couramment utilisés est celui des sciences de l'information géographique.

L'entrepôt de données (*« data warehouse »*) est au sens large le stockage centralisé d'un grand volume de données factuelles dans une base de données. La structure est optimisée pour l'interrogation et l'analyse. Dans le contexte des données d'entreprise, cette structure repose souvent sur une approche multidimensionnelle. On stocke généralement l'historique de toutes les données d'une entreprise pour l'aide à la décision [Inmon 1992; Kimball and Ross 1996]. Il existe aujourd'hui des architectures qui proposent l'instanciation de portions de l'entrepôt dans des contextes plus spécifiques appelés *« data marts »* :

> *The Data Warehouse is a subject-oriented, integrated, time-variant, non-volatile collection of data used to support the strategic decision-making process for the enterprise. It is the central point of data integration for business intelligence and is the source of data for the data marts, delivering a common view of enterprise data* [1]*[Inmon 1992]*

L'intégration est alors abordée par Bill Inmon comme la réunion des données à partir de sources distribuées et hétérogènes dans un tout cohérent. Il existe principalement deux écoles. Selon l'approche descendante d'Inmon, l'entrepôt est une partie du système de décision, les données des *« marts »* provenant de l'entrepôt [Inmon 1992]. Ralph Kimball oppose une approche ascendante, l'entrepôt se construit par l'agglomération des sources locales [Kimball and Ross 1996].

Dans notre contexte bioinformatique, l'intégration est un besoin qui découle de l'existence de sources hétérogènes. Cette notion est plus proche des propositions de Kimball. La matérialisation des données procure de multiples avantages motivant son choix :

- les performances sont nettement supérieures,
- il est possible de mettre en œuvre une historisation, des métadonnées, et des statistiques sur les usages,
- la communauté d'utilisateurs est propriétaire des données, il est possible de les modifier dans le but d'un nettoyage,
- la confidentialité et la sécurité sont des considérations critiques et légales dans le domaine médical. Dans le domaine biomédical, les enjeux financiers motivent aussi une confidentialité des travaux. Lors de l'interrogation de systèmes distants, l'utilisateur craint qu'il n'y ait une trace ou une mémorisation de ses requêtes en vue d'un espionnage industriel. Il préfère souvent télécharger l'ensemble des données d'une source et travailler localement avec afin de s'assurer que personne ne puisse prédire les directions actuelles de l'entreprise ou de l'équipe de recherche.

L'historisation et la présence d'un modèle sont deux fonctionnalités essentielles de l'entrepôt. L'historisation consiste dans un premier temps à dater les informations. Elle ne suffit pas.

[1] L'entrepôt de données est une collection de données « métier », intégrées, évolutives, historisées et non volatiles qui supporte le processus de prise de décision stratégique dans l'entreprise. Il est l'élément central de l'intégration de données pour le décisionnel et la source de données pour les « *data marts* », fournissant une vue commune des données de l'entreprise.

[Buneman, Khanna et al. 2001] discutent de la *« why provenance »* et la *« where provenance »* trop souvent négligées : D'où viennent les données ? Pourquoi et comment ont-elles été intégrées ? Quand un système ne possède ni historisation ni structure rigoureuse, on parle généralement de dépôt de données (*« data repository »*). On peut par exemple citer Ligand Depot [Feng, Chen et al. 2004] et PePr Warehouse [Chen, Zhao et al. 2004].

Les deux limites de cette approche résident dans sa lourdeur. La première concerne la mise en œuvre ; l'entrepôt nécessite généralement un système matériel et logiciel plus coûteux. Le second problème, majeur, concerne la maintenance à jour des données. La mise à jour dépend de deux paramètres : *« When »* et *« How »*, quand et comment ? [Engström and Asthorsson 2003]. Mettre à jour implique dans un premier temps de détecter que les données sources ont été modifiées, et quelle partie de ces données a été mise à jour. Concernant la mise à jour des données différentes solutions sont proposées [Gupta and Mumick 1995]. Dans un second temps, l'évolution des données peut rendre obsolètes les résultats d'une fouille. Dans ce cas, il faut exécuter à nouveau les procédures de fouille. L'idéal est alors de disposer de procédures incrémentales. Le problème de l'intégration se complique lorsque les sources voient leur schéma évoluer [Chen, Chen et al. 2004], cependant, ce problème est aussi présent dans le cas d'une approche médiateur.

On considère qu'il existe différents niveaux d'intégration dans l'entrepôt [Davidson, Overton et al. 1995]. Pour construire le schéma global, la solution la plus simple est de choisir l'union des schémas des sources locales. L'avantage est la simplicité d'intégration des données à partir des sources. L'hétérogénéité des schémas est gérée durant la mise en œuvre des requêtes. Elle nécessite la spécification d'alignements entre les différentes sous-parties du schéma global (*« schema mappings »*). L'intégration syntaxique rend plus complexe l'intégration des instances (verticale).

L'intégration sémantique propose un schéma global unifié, modélisateur. L'interrogation d'un schéma unique est plus facile, la difficulté réside dans la conception des adaptateurs (*« wrappers »*) qui traduisent les données depuis les schémas locaux vers le schéma global. Cette intégration des schémas (horizontale) facilite l'intégration des instances (verticale).

Exemples

On distingue différentes approches entrepôts correspondant aux différents modèles de bases de données. Les exemples qui suivent concernent uniquement le domaine biomédical. **Acnuc** [Gouy, Milleret et al. 1984] est l'un des premiers systèmes. Il est basé sur le modèle réseau. **GeWare**, se focalise sur les données d'expression et adopte un modèle multidimensionnel [Rahm, Kirsten et al. 2007].

GUS (*Genomics Unified Schema*) est actuellement implanté dans plus d'une vingtaine de projets [Davidson, Crabtree et al. 2001]. Il repose sur le paradigme relationnel avec une surcouche objet fournie par Oracle. GUS possède une intégration syntaxique (le schéma global est l'union des schémas locaux). Une partie de son schéma, cependant, le « *core* » permet de conserver une traçabilité sur les algorithmes et leur mise en œuvre, en particulier sur l'intégration verticale. De plus, les sous-ensembles du schéma sont des schémas médiateurs ou correspondant à des normes. Ils ne proviennent pas directement d'une source unique. Pour ces raisons, on considère souvent GUS comme un entrepôt intermédiaire entre le niveau syntaxique et sémantique. Notons enfin qu'il est fourni avec un portail en ligne proposant de nombreux services (un exemple de page est proposé dans l'annexe C.1.1 (294).

EnsMart – EnsMart est un entrepôt relationnel qui intègre les données d'Ensembl et d'autres bases de données reliées [Kasprzyk, Keefe et al. 2004]. L'intégration se déroule en deux étapes. La première récupère les données externes et les charge dans le SGBDR en utilisant les schémas locaux. La seconde intègre dans le schéma global chaque base de données intermédiaire en utilisant des procédures spécifiquement développées (en Perl). Des interfaces et API et langage de script sont disponibles.

SeqHound – SeqHound est un entrepôt relationnel construit pour stocker les ressources du NCBI [Michalickova, Bader et al. 2002]. Les auteurs évoquent les limites abordées précédemment liées au portail d'Entrez : problème de performance pour la fouille, de traçabilité et propriété des données, d'accès via un langage de requête évolué, etc. Chaque tuple correspond à un fichier d'Entrez. Il s'agit donc plus d'une banque de données ou d'un dépôt de données que d'un système d'intégration.

UMLS est un entrepôt intégrant une centaine d'ontologies. Nous l'avons déjà décrit dans la section 2.3.1.2 (page 55) [Bodenreider 2004]. Les données sont décomposées en attributs atomiques dans un schéma relationnel et interrogeables en SQL. Un portail en ligne est proposé (UMLSKS) et une application locale permet d'interroger les fichiers sans SGBDR. L'intégration sémantique est horizontale (présence d'un schéma global) et verticale (Metathesaurus qui sépare concepts et atomes). Notons que le schéma relationnel ne respecte pas la seconde forme normale de Codd.

Gedaw est un entrepôt orienté objet dédié à l'annotation fonctionnelle à partir de données d'expression du transcriptome hépatique [Guérin, Marquet et al. 2005]. Il met en œuvre une intégration sémantique horizontale et verticale par des règles de correspondance. Son principal atout est la qualité des données contenues. L'intégration du schéma se fait par la détection d'analogies structurelles. Les sources intégrées sont cependant limitées aux formats structurés (relationnel : UMLS, GO, etc.) et semi-structurés (XML : GenBank, etc.). Une intégration verticale est aussi réalisée au travers de règles de correspondance donnant lieu à des regroupements. Ces règles sont éditables par un expert.

GIMS est un entrepôt orienté objet dédié à l'annotation du génome [Cornell, Paton et al. 2001] qui a notamment l'originalité de mettre à disposition des outils graphiques pour spécifier une requête et l'analyser. Le modèle objet est utilisé comme point d'entrée pour la navigation et la construction de la requête.

AceDB (a *C. elegans* database) est un système de gestion de base de données orienté objet mais dédié à l'intégration de données génomiques [Stein, Sternberg et al. 2001]. Il a été initialement conçu pour un projet de séquençage de l'organisme *Caenorhabditis Elegans*, un ver commun qui mesure 1 mm de long. Aujourd'hui, le portail s'appelle *Wormbase*. L'objectif principal de ce projet est de fournir un système améliorant l'évolutivité et l'extensibilité du schéma. Ce système est devenu un système d'intégration complet comme en témoignent plus de cinquante bases de données qui reposent dessus. Il propose par ailleurs des outils graphiques moins souples et adaptables que le SGBD lui-même.

2.3.2.1 L'approche médiateur (vues virtuelles)

Définition et caractéristiques

Le médiateur, dans son acception la plus générale, est une entité qui facilite la communication entre plusieurs agents. Il facilite la négociation et la résolution de conflits. C'est un intermédiaire qui contrôle le bon déroulement de la communication, et peut imposer un protocole sur le média. En informatique, le médiateur est issu de l'apparition de bases de données distribuées et hétérogènes, il permet la communication entre plusieurs systèmes par le partage d'un schéma global commun. Plus précisément, sa signification en informatique est celle d'un système d'intégration qui ne suit pas une approche matérialisée :

> *A mediator is a software module that exploits encoded knowledge about some sets or subsets of data to create information for a higher layer of applications*[1] *[Wiederhold 1992].*

Le système d'intégration permet d'interroger plusieurs sources de façon unifiée via un langage de requête. Les entrepôts inventoriés précédemment disposent immédiatement d'un tel

[1] *« Un médiateur est un module logiciel qui exploite la connaissance de certains ensembles ou sous-ensembles de données pour créer de l'information pour des applications à un niveau supérieur. »*

outil puisqu'ils sont matérialisés dans une base de données. Le médiateur au contraire ne stocke pas les données dans un espace central. Il permet cependant d'accéder aux sources au travers de vues virtuelles. Nous avons ainsi retenu la définition de S. Cohen-Boulakia :

> *Un médiateur donne à l'utilisateur l'illusion d'interroger un système homogène et centralisé [Cohen-Boulakia 2005]*

Cette approche, issue des bases de données fédérées, repose sur le principe des vues. Les données ne sont pas matérialisées, contrairement à un entrepôt, d'où le nom d'intégration *virtuelle*. L'interrogation se fait au travers d'un schéma pivot. A la différence de l'entrepôt, cette approche nécessite généralement l'établissement d'un plan de requêtes : il faut identifier les sources nécessaires, traduire les requêtes vers les schémas locaux, puis expédier ces requêtes. Les résultats sont alors rapatriés puis fusionnés. Cette fusion est comparable à l'intégration dans l'entrepôt : il faut traduire le résultat dans un schéma commun (intégration horizontale) puis éliminer la redondance et établir des correspondances (intégration verticale).

L'avantage immédiat de cette approche, comme pour la vue, est l'absence de coût d'actualisation des données. De plus, cela ne demande pas d'investissement préalable (pas de matériel à dédier, aucun serveur à administrer, etc.). On considère ainsi cette approche comme plus légère à mettre en œuvre. Plusieurs inconvénients subsistent :

- il est difficile de concevoir des adaptateurs, l'intégration verticale peut aussi s'avérer plus complexe à mettre en œuvre.
- l'utilisateur n'est pas propriétaire des données : il est impossible de les modifier, nettoyer, etc.
- les performances sont limitées par celles des sources et du réseau.
- la complétude du résultat est soumise à la disponibilité de toutes les sources au moment de l'interrogation. Plus il y a de sources, plus il y a de risque d'incomplétude.
- l'utilisateur n'a pas accès aux métadonnées des sources et n'a aucune information et aucun contrôle sur l'historique et les versions des données.
- dans le cas d'un entrepôt, si le schéma d'une source évolue, cela représente une limite en termes d'actualité des données. Dans le cas d'une approche virtuelle, c'est l'accès total aux données qui est interrompu.
- dans une approche non matérialisée, il est plus difficile et coûteux d'établir des contraintes et jointures entre les sources [Davidson, Overton et al. 1995].

Au delà de ces contraintes fréquemment exprimées, un problème persiste, qui provient de l'hypothèse sur laquelle repose le principe du médiateur : les sources sont accessibles par un mécanisme de requêtes structurées. Or, actuellement, seule une minorité de systèmes sont interrogeables librement par un langage comme SQL ou OQL.

Wiederhold définit trois couches dans la médiation :

- la couche de base dans laquelle s'inscrivent les bases de données et les différents outils de fouille, de calcul, etc.
- la couche médiation qui contient différents médiateurs. Ces médiateurs sont spécifiques à des domaines de spécialité, proposent des services à valeur ajoutée.
- La couche application dans laquelle se situent les outils directement manipulés par l'utilisateur pour sa prise de décision.

Wiederhold souligne l'importance de la distinction entre le couche médiation et les deux autres. Dans cette approche, on peut comparer le médiateur au contrôleur de dialogue dans les architectures d'applications. On distingue plusieurs approches dans la construction du schéma global.

L'approche *« Global As View »* (GAV) consiste à construire le schéma global en fonction des schémas locaux. Cela nécessite que les sources soient connues *a priori*. L'avantage est qu'une adéquation entre le schéma global et les schémas locaux facilite la réécriture de requêtes. L'inconvénient est que l'ajout ou l'évolution du schéma de sources *a posteriori* remet en cause le schéma global.

L'approche opposée, *« Local As View »* (LAV) consiste à construire un schéma global modélisateur du domaine. Par la suite, chaque source locale s'adapte à ce schéma. Cette approche gagne en souplesse vis-à-vis de l'ajout ou de l'évolution des sources. Mais l'incohérence entre le schéma global et certains schémas locaux peut rendre difficile la conception des adaptateurs pour la réécriture des requêtes et la fusion des données [Levy 1999] : certaines entités peuvent ne pas être représentables dans le schéma global.

Enfin, il existe une approche mixte *« Generalized Local As View »* (GLAV) qui construit des vues au niveau local et global [Friedman, Levy et al. 1999; Calvanese, De Giacomo et al. 2001]. Cela permet d'obtenir un compromis entre souplesse pour l'ajout des sources et palliation des difficultés de réécriture des requêtes et de fusion des résultats. Cette approche accentue la décentralisation et se rapproche des bases de données fédérées. Elle est notamment incarnée par les systèmes *« pair à pair »*, multi-agents ou grilles.

Exemples

Discovery Link est un système commercial conçu par IBM [Haas, Schwarz et al. 2001]. Il repose sur un modèle relationnel-objet et sur le serveur de la même organisation, DB2. Il permet d'interroger une base de données virtuelle directement en SQL. Il est notamment reconnu pour ses excellentes performances. En effet, il propose un système d'optimisation efficace basé sur les coûts et sur une mémoire cache qui conserve localement une partie des données. L'intégration est syntaxique : dans un premier temps, les données des sources locales sont traduites par des algorithmes dans le paradigme relationnel exploitable par le système (cette procédure s'apparente à une approche GLAV). Par la suite, on établit des correspondances dans les adaptateurs entre le schéma relationnel généré et le schéma global. Actuellement, Discovery Link évolue vers la gamme de produits du nom de WebSphere Information Integrator.

SEMEDA (SEmantic MEta DAtabase) est un médiateur qui a la particularité de se baser sur une (ou plusieurs) ontologie(s) [Kohler and Schulze-Kremer 2002; Köhler, Philippi et al. 2003]. Un SBGDR assure le stockage des métadonnées. Semeda se définit même comme un système pour intégrer des données ou concevoir et maintenir collaborativement des ontologies.

TSIMMIS *(The Stanford-IBM Manager of Multiple Information Sources)* est un des premiers systèmes médiateurs orienté objet [Chawathe, Garcia-Molina et al. 1994]. Il résulte d'une collaboration entre l'université de Stanford et IBM. Il repose sur le paradigme objet, utilisant OEM (Object Exchange Model) pour spécifier le schéma et OEM-QL comme langage de requête. Il suit une approche GAV. **K2** (anciennement **Kleisli**) est le système développé par l'université de Pennsylvanie [Davidson, Crabtree et al. 2001]. Il repose sur une approche GAV. Ses atouts sont :

- un langage déclaratif de haut niveau, K2ML pour spécifier les transformations de schémas,
- un mécanisme de polymorphisme de types offrant plus de souplesse vis-à-vis de l'évolution des sources,
- un optimiseur de requêtes qui permet de stocker les résultats (matérialisation)
- la possibilité de réaliser des jointures entre les sources.

D'autres systèmes utilisent des approches à base de règles ou de logique. Concernant les données biologiques, les deux projets les plus connus sont Tambis et Baciis. **Tambis** (*« Transparent Access to Multiple Biological Information Source »*) possède une ontologie TaO (approche GAV) pour l'intégration [Baker, Brass et al. 1998]. Cette ontologie est utilisée à la fois pour l'intégration des schémas mais aussi comme vocabulaire contrôlé pour l'intégration même des termes employés dans les sources locales. Tambis exploite le langage CPL de Kleisli pour la spécification des adaptateurs. **Baciis** (*« Biological and chemical information integration system »*) repose aussi sur une ontologie BAO (approche GAV). On retrouve enfin quelques expériences exploitant les logiques modales et contextuelles [Farquhar, Dappert et al. 1995].

2.3.3 Systèmes à base de liens et chemins

2.3.3.1 Systèmes à base de liens

Présentation

Les approches précédentes sont appelées systèmes d'intégration : elles fournissent une vue unifiée sur les données et donne l'impression pour le développeur-utilisateur de manipuler un SGBD centralisé. Le système est interrogé directement au travers du langage de requête (SQL, OQL, etc.), qu'il s'agisse d'une approche médiateur ou entrepôt, le schéma est perçu comme une base de données.

« One approach to providing a useful query interface to a collection of data sources is to create a layer of links between database records on top of the data sources. » [Bökman 2001]

Une autre approche de l'intégration de données consiste à proposer une interface utilisateur commune et unique. Au lieu de s'appuyer sur un schéma et une vision atomique des données, ceux-ci considèrent les données au niveau du document ou du fichier et s'intéressent uniquement aux liens entre ces documents (références croisées). Ces systèmes sons regroupés sous le terme « systèmes à base de liens » (ou *« médiation orientée liens »*[1][Davidson, Crabtree et al. 2001]), mais certains parlent parfois plus simplement de *portails* ou *navigateurs* [Cohen-Boulakia 2005]. Les premiers systèmes d'information biologiques étaient pour l'essentiel de simples dépôts de données primaires contenant des fichiers et non des bases de données. La nature même du Web est l'hypertexte, la navigation par des liens entre les documents. C'est donc assez naturellement que les premiers systèmes d'intégrations ayant émergé ont été des systèmes à base de liens. Ces systèmes ont cependant fortement évolué et proposent maintenant de nombreux autres services.

Exemples

Acnuc est historiquement l'un des premiers [Gouy, Milleret et al. 1984]. Il s'agit en fait d'un SGBD organisé en réseau/hiérarchies. Il est assorti d'un portail utilisateur (en tant qu'interface en ligne). Parfois classé comme un portail [Cohen-Boulakia 2005], il est le plus souvent apparenté à un entrepôt : les données sont matérialisées au sein d'une base de données possédant un schéma et sont interrogeables via un langage de requête. De plus, il ne se focalise pas sur les références croisées.

Entrez est le portail du NCBI, probablement le plus utilisé des portails. Des références croisées sont ajoutées aux données ainsi que des relations de voisinage. Ces relations sont calculées automatiquement. Il s'agit essentiellement pour les séquences de recherche de similarité (Blast). Les données ne sont pas matérialisées, les références croisées et les relations de voisinage précalculées le sont. Du point de vue de l'interface utilisateur en ligne, Entrez dispose d'un mécanisme de requête avancé et propose des comptes utilisateurs : MyNCBI (introduit en 2006). Concernant le mécanisme de requête, les mots clés peuvent contenir des *jokers* (par exemple *« promyelo* »* signifie *tous les mots commençant par « promyelo »*). Il est possible de spécifier des champs sur lesquels s'appliquent spécifiquement la recherche (organisme, titre, résumé, etc.) et de les articuler à l'aide d'opérateurs booléens. Il est aussi possible de spécifier un ensemble de filtres : type de publication, accès au contenu intégral, dates, sujets, etc. Les comptes utilisateurs permettent de conserver un historique des requêtes. Il est ainsi possible de combiner des requêtes, (raffinements), et de « stocker » leurs résultats dans des collections. Par la suite, ces collections peuvent être analysées, éditées manuellement, mises à jour, fusionnées ; il est également possible de trier les résultats ou encore d'activer des mécanismes d'alerte. Des utilitaires permettent l'accès sous forme de services à Entrez. On peut

[1] *« link driven federation »*

aussi, plus simplement, l'interroger au travers d'une requête HTTP. Ces requêtes ne doivent cependant pas être exécutées avec une fréquence importante.

Expasy (Expert Protein Analysis System) est le portail suisse spécialisé dans les protéines, leurs séquences et leurs annotations [Gasteiger, Gattiker et al. 2003]. Il propose des références croisées vers plus de 70 ressources externes, 8 en moyenne pour chaque protéine en 2003. Il importe lui-même 5 sources. Les mécanismes de recherche sont initialement assez simples : mots clés, spécification de quelques attributs, recherche « plein texte ». Ils ont été complétés par des mécanismes d'alerte et par l'utilisation du portail SRS (décrit plus loin). Il propose de nombreux services de traduction, d'analyse et de visualisation des séquences, et des outils protéomiques divers. A nouveau, certaines relations sont précalculées et stockées.

GeneCards est un outil qui stocke les données localement [Rebhan, Chalifa-Caspi et al. 1998; Safran, Solomon et al. 2002]. Régulièrement, il recharge ses données à partir des *« dumps »* en les convertissant dans un format plat spécifique (une *GeneCard*). Des algorithmes d'extraction génèrent alors les hyperliens entre les fiches. Il utilise Glimpse pour la recherche « plein texte » et Excite pour une recherche indexée. Un effort particulier est affiché pour exploiter et respecter la nomenclature des gènes HUGO [Eyre, Ducluzeau et al. 2006].

LinkDB/DBGet est le portail japonais de GenomeNet de l'Université de Kyoto, qui met notamment à disposition la source KEGG [Fujibuchi, Goto et al. 1998]. Il contient des fichiers « plats » et propose quatre types de liens : les liens de références croisées (qui sont complétés par les liens réciproques), les liens de similarités introduits par Entrez, et les liens « biologiques » correspondant aux réactions biochimiques des voies métaboliques ou de signalisation. LinkDB, contrairement à SRS, a fait le choix de ne pas fournir de langage de requête permettant de spécifier des chemins, mais de précalculer certains chemins ajoutés et de générer des arêtes directes. Cela procure trois avantages :

- on gagne en performances et coûts de calculs,
- l'utilisateur n'a pas besoin de connaître les sources et de spécifier des chemins, les liens indirects pertinents sont déjà présents,
- lors de l'ajout d'une source et dans le cadre d'une historisation des requêtes, il est possible d'alerter l'utilisateur des nouvelles relations établies.

2.3.3.2 Systèmes à base de chemins

Présentation

> *Public biological resources form a complex maze of heterogeneous data sources, interconnected with links and applications. This valuable network offers scientists potential answers to a wide variety of scientific questions. […] While scientific questions are posed at a conceptual level, their implementation entails determining which data resources and tools to use, that is, which **paths** in the network of sources to follow. [Cohen-Boulakia, Davidson et al. 2006]*[1]

On distingue généralement les approches basées sur les liens des approches basées sur des scénarios. Ces approches s'orientent vers la notion de plateforme, que Sarah Cohen-Boulakia définit comme apportant un support à l'analyse [Cohen-Boulakia 2005]. Dans [Cohen-Boulakia, Davidson et al. 2006], elle précise qu'un scénario peut être considéré comme l'expression d'un protocole scientifique *in silico*. De façon générale, le biologiste qui navigue sur des bases de

[1] *Les ressources biologiques publiques forment un labyrinthe complexe de sources de données hétérogènes, interconnectées avec des liens et des applications. Ce réseau valable offre aux scientifiques des réponses potentielles à une large variété de questions scientifiques. […] Tandis que les questions scientifiques sont posées à un niveau conceptuel, leur implémentation nécessite de déterminer quelles sources de données et outils utiliser, c'est-à-dire, quels chemins suivre dans le réseau de sources. [Cohen-Boulakia, Davidson et al. 2006]*

données partagées en ligne spécifie des mots clés, suit des liens, applique des filtres et des critères de tri : c'est un scénario. Dans le contexte d'une approche à base de liens, le scénario est défini comme un chemin dans le graphe. Par exemple, répondre à la question *« Quels sont les gènes humains impliqués dans la leucémie ? »* correspond à :

- rechercher l'ensemble des chemins reliant : les organismes, leurs gènes, et les maladies,
- restreindre le résultat à la valeur *« homo sapiens »* pour l'organisme, et « leucémie » pour l'affection,
- retourner le résultat à l'utilisateur.

Le système à base de liens SRS présenté dans la suite est l'exemple d'une approche à base de chemins intermédiaire entre le portail et la plateforme. Les systèmes présentés par la suite sont de réelles plateformes qui reposent sur une représentation de graphe. Plus précisément, ces approches reposent sur la représentation de deux graphes : le graphe conceptuel[1] qui représente les entités biologiques et leurs relations, et le graphe physique qui représente les sources, leurs entités, leurs attributs, et les relations associées (outils, références croisées, etc.). Ces différents exemples sont présentés dans la suite de cette section ; nous nous appuyons sur l'analyse comparative présentée dans [Cohen-Boulakia, Davidson et al. 2006] à laquelle le lecteur peut se référer pour de plus amples informations.

Exemples

SRS (Sequence Retrieval System) a déjà été mentionné dans la section précédente [Etzold, Ulyanov et al. 1996; Zdobnov, Lopez et al. 2002]. Il s'agit en effet d'un des premiers systèmes du genre qui, de plus, s'avère remarquable. A l'instar d'Entrez et de ses confrères, SRS propose d'ajouter des hyperliens à partir de données contenues dans les fichiers à plat. Une évolution dans la version 6 permet de découper une entrée (fichier) en sous-entrées (domaine, commentaire, référence bibliographique, etc.).

Les données ne sont pas obligatoirement stockées, alors que les références croisées sont stockées. Cependant, il est aussi possible de stocker les données localement pour accroître les performances. Il faut noter qu'actuellement, les données sont issues de plus de 400 sources. Cette extensibilité est notamment permise grâce au langage Icarus qui permet d'écrire des adaptateurs rapidement. Ce langage est interprété et possède une syntaxe proche de Perl.

L'interface utilisateur permet de naviguer entre les sources, comme les autres systèmes, mais propose aussi un langage puissant pour interroger les données : Getz. Ce langage permet d'exprimer des chemins entre les sources, d'appliquer des filtres, services, etc. Les accès peuvent se faire au travers du portail en ligne, d'un client en ligne de commande ou en l'encapsulant dans une URL. Les résultats peuvent être temporairement stockés puis réutilisés. Les capacités de cet environnement le rapprochent d'une plateforme, comme le montre son utilisation au sein de BioGuideSRS (cf. plus loin dans cette section)[Cohen-Boulakia 2005; Cohen-Boulakia, Davidson et al. 2006]. Plus généralement, SRS est interrogeable via CORBA et une API disponible en Java, Perl, Python et C++. La limite reste principalement que l'utilisateur doit connaître les nombreuses sources, et leur structure pour manipuler efficacement cet outil.

BioMediator est le premier système de ce type à reposer sur une représentation en graphe [Donelson, Tarczy-Hornoch et al. 2003]. Comme son nom l'indique, il suit une approche non matérialisée : les liens sont générés à la volée. Les chemins exprimés sont donc définis comme des plans de requête. Il repose sur une approche XML, les requêtes sont spécifiées en XQuery. Les résultats sont visualisés à l'aide de la boîte à outils TouchGraph (cf. figure 2.15).

[1] Il est défini comme tel par [Cohen-Boulakia, Davidson et al. 2006], cette définition n'a aucun rapport avec les graphes conceptuels de Sowa.

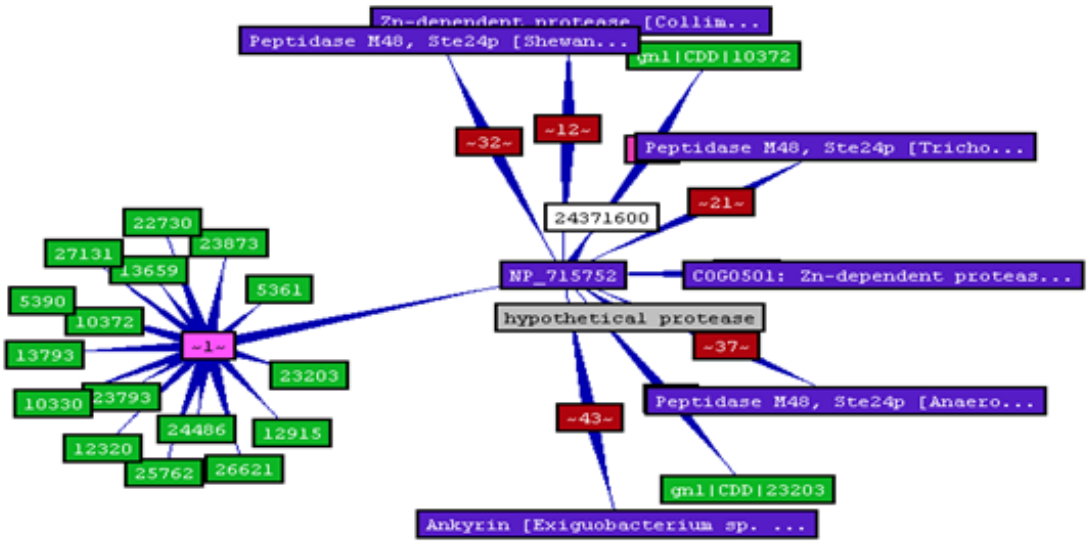


Figure 2.15 – BioMediator utilise TouchGraph pour visualiser le graphe résultat (visualisation basée sur un modèle physique.

Biozon est un projet similaire [Birkland and Yona 2006]. La spécification de la requête s'effectue au travers d'un formulaire en ligne. Le graphe conceptuel se limite à huit entités biologiques et une seule relation est autorisée entre deux entités. Contrairement à BioMediator, les données sont matérialisées dans un entrepôt.

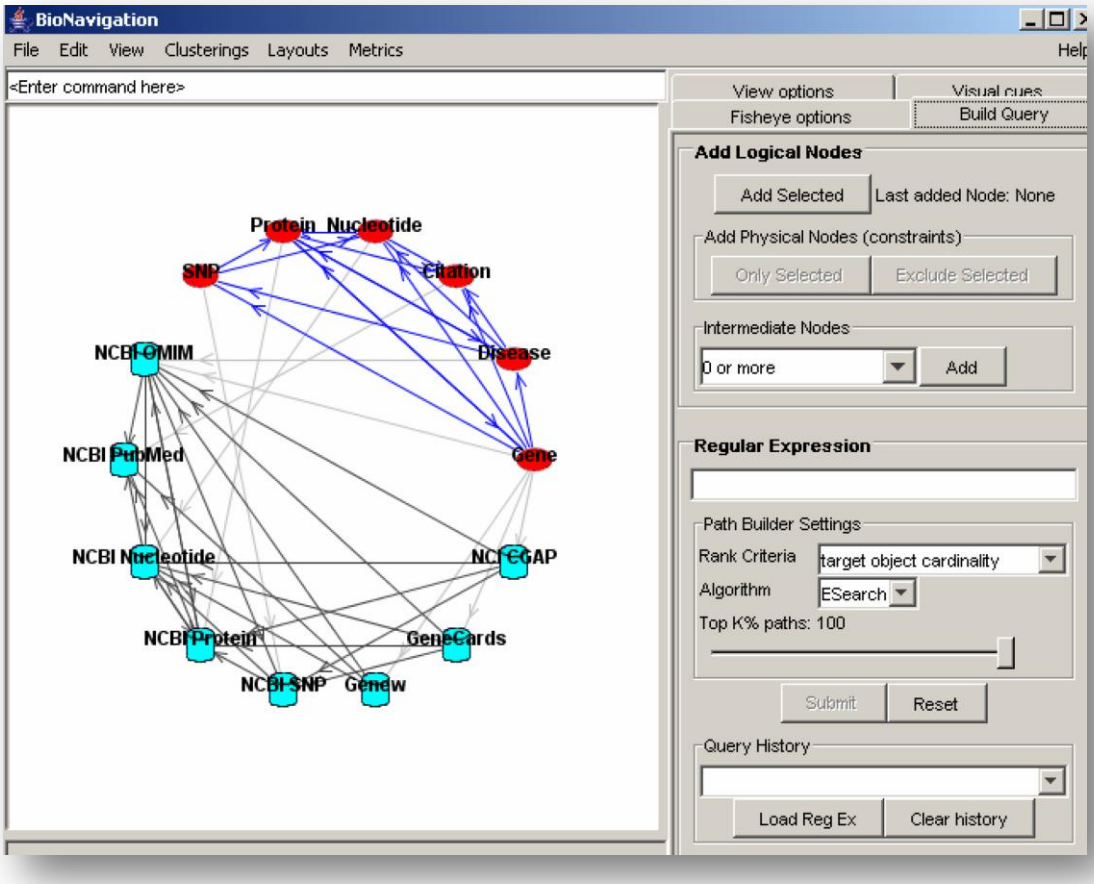


Figure 2.16 – Captures d'écran de BioNavigation. Le haut du graphe (en rouge) représente le graphe conceptuel (entités), le bas (en bleu) représente le graphe physique (sources).

BioNavigaton se distingue des deux systèmes précédents [Lacroix, Parekh et al. 2005]. Il propose une manipulation du graphe physique, une fonction absente dans les systèmes précédents. Cependant, seule une relation peut exister entre deux sources. Les requêtes correspondent à des expressions régulières et peuvent contenir des étoiles « * » en tant que fermeture de Kleene. Par exemple, *« Gène.*.Maladie(Leucémie) »* répondrait à la question : *« Quels sont les gènes en relation avec la leucémie ? »* Les requêtes sont spécifiées graphiquement. Dans la capture d'écran suivante (figure 2.16), la partie gauche présente les deux graphes, conceptuels et physiques. La partie droite permet de spécifier la requête. Cette requête peut s'exprimer en quelques clics ou sous forme d'une expression régulière dans le champ de texte visible dans la capture. Le résultat de la requête est une liste ordonnée de chemins élémentaires[1]. Cette liste est obtenue à l'aide de l'algorithme ESearch.

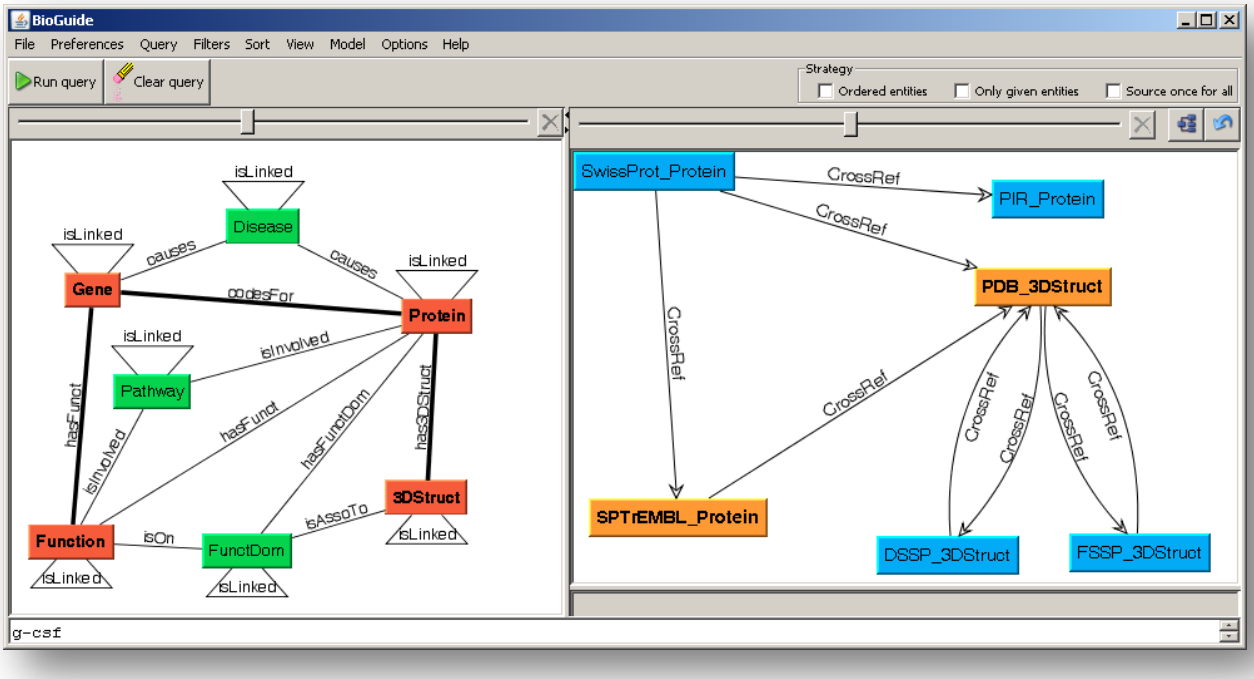


Figure 2.17 – Captures d'écran de BioGuide. Dans les deux graphes, les entités et relations qui sont en gras/épaissies sont sélectionnées. La partie gauche montre le graphe des entités. Le graphe des sources associées est présenté dans la partie droite de la fenêtre.

BioGuide comme les environnements précédents repose sur un modèle de graphe (conceptuel et physique) et permet de représenter des chemins [Cohen-Boulakia 2005; Cohen-Boulakia, Biton et al. 2007]. Il apporte plusieurs bénéfices vis-à-vis de BioNavigation.

1. Il permet de représenter plusieurs liens entre deux sources. La spécification de la requête est graphique et via un champ de texte situé en bas de la fenêtre (figure 2.17).
2. Les arêtes sont étiquetées.
3. Une source peut être reliée à plusieurs entités.

[1] Sans circuit

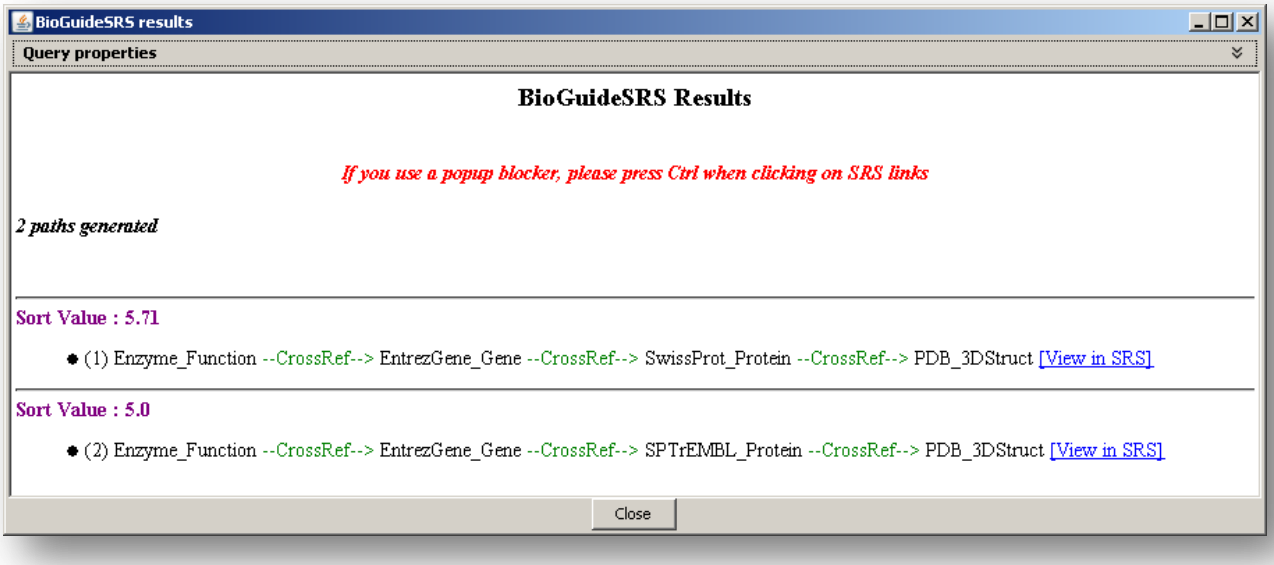


Figure 2.18 – Exemple de résultat dans BioGuide : on constate l'affichage de deux chemins, pondérés.

Un exemple de résultat obtenu est illustré dans la figure 2.18. De plus, de nombreux efforts sont mis en œuvre concernant la spécification et l'exploitation de préférences à trois niveaux : global (longueur maximum d'un chemin), intermédiaire (qualité de certaines sources et relations) ou local (choisir ou éviter une source particulière). BioGuide repose sur un langage de requête spécifique (XPR – eXtensible Path language for RDF). BioGuide est indépendant d'une plateforme. Il a été développé initialement par le LRI et l'Université de Pennsylvanie associés à un partenaire privé, ISoft qui disposait d'un environnement de flux de travail, Amadea. Par la suite, BioGuide a été porté en quelques heures vers SRS. Il a suffi de décrire le métamodèle de SRS, les requêtes sont alors traduites de BioGuide vers la syntaxe de Getz.

Une autre limite de BioNavigation vis-à-vis de BioGuide est la contrainte obligeant le parcours de chemins élémentaires dans le graphe des sources. Cette contrainte est mise en œuvre afin de pallier l'explosion combinatoire. Cependant, elle induit des limites d'usage. Prenons l'exemple suivant[1] : Un chemin résultat d'une requête traverse 6 fiches issues de deux sources GenBank et SwissProt :

$$SP_1 \rightarrow GB_1 \rightarrow GB_2 \rightarrow GB_3 \rightarrow GB_4 \rightarrow SP_2.$$

Ce chemin est long (5 liens) et boucle vers la source initiale. Deux cas sont alors possibles : SP_1 et SP_2 sont identiques ou différents. Le premier cas peut s'avérer utile afin de vérifier la précision du résultat. D'un cas à l'autre, les références croisées peuvent s'avérer nombreuses, et le facteur de branchement peut générer un grand nombre de chemins qui n'ont plus qu'un rapport éloigné avec l'élément initial. Le problème est d'autant plus important que le chemin est long. Dans cet exemple, la contrainte $SP_1 = SP_2$ permet de s'assurer que GB_3 ou GB_4 est toujours « proche » de SP_1. Cela permet de garantir une précision et une fiabilité dans le résultat qu'il n'y a pas eu de dispersion. Dans le cas où les éléments sont différents, il peut tout simplement être pertinent d'analyser SP_2, notamment si le chemin est assez court.

Plus généralement, [Cohen-Boulakia, Davidson et al. 2006] différencie BioNavigation et BioGuide par des objectifs distincts : BioNavigation recherche une efficacité algorithmique pour fournir rapidement un chemin le plus court. La recherche par une expression régulière en utilisant l'étoile de Kleene peut, en effet, aboutir à un nombre exponentiel de chemins. BioGuide

[1] Je remercie par ailleurs Sarah pour les explications complémentaires qu'elle m'a apportées. Cet exemple est issu de l'une de nos discussions.

recherche au contraire un ensemble de chemins exhaustif, mais plus restreint en raison du système de requête. L'obtention en ordre pertinent en accord avec les préférences utilisateur est une priorité, et la gestion de ces préférences est complète.

GenoLink [Durand, Labarre et al. 2006] est produit par le consortium GenoStar, qui réunit quatre partenaires, publics (INRIA et Institut Pasteur) et privés (Genome Express, Hybrigenics). C'est un système sensiblement différent : le pouvoir d'expression est alors bien supérieur, les requêtes ne sont pas limitées à des chemins, les résultats non plus. Les requêtes sont spécifiées interactivement (figure 2.19) et les résultats bénéficient de visualisations.

Ce système est assimilé à un outil de fouille de graphe. Le problème d'un point de vue algorithmique est voisin de celui de la recherche d'un isomorphisme dans un graphe qui est NP-complet. GenoLink repose sur un entrepôt utilisant AROM comme langage de modélisation [Page, Gensel et al. 2000], et sur une API. Un langage de requête spécifique est mis en œuvre : GQL (GenoLink Query Language), qui s'apparente à un langage de script.

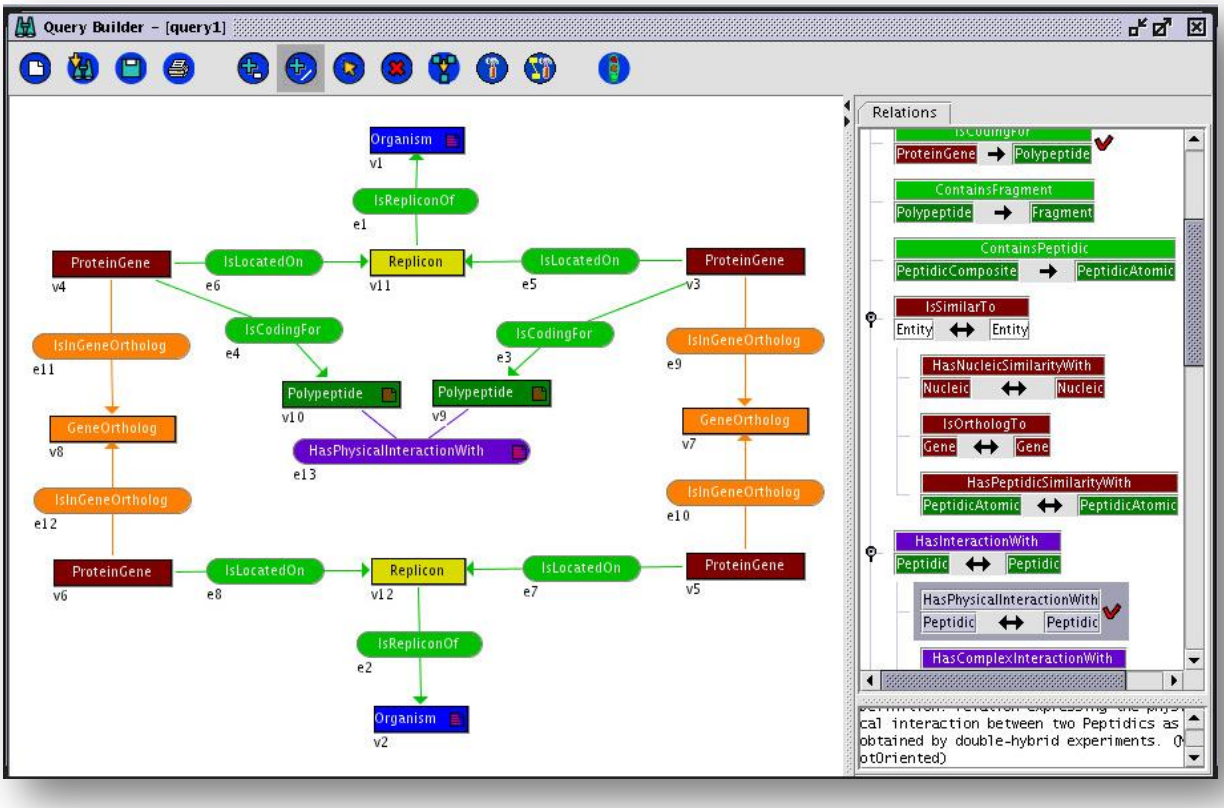


Figure 2.19 – Exemple d'une requête dans GenoLink. La partie gauche contient la requête. Les boîtes carrées (foncées) sont des entités, les boîtes arrondies (plus claires) représentent les étiquettes des relations. La partie droite propose une vue du métamodèle pour construire la requête.
A partir d'une interaction observée entre deux protéines dans un organisme (v1, en haut du graphe), on cherche à inférer des interactions entre deux protéines similaires dans un autre organisme (orthologues).

2.3.4 Plateformes et environnement intégrés

Une plateforme intégrée ou un environnement intégré est un support pour l'analyse et permet de mettre en œuvre des services, outils, calculs, etc. Le nombre d'outils de ce type est particulièrement important en bioinformatique. Ils suivent différentes approches. On peut opposer trois axes :

- Des **initiatives isolées** sont dédiées à une ou plusieurs utilisations très spécifiques. Imagene s'intéresse ainsi à l'analyse de puces à ADN [Médigue, Rechenmann et al. 1999]. Il intègre des outils d'analyse d'image, d'analyse de données, et interroge GenBank, EMBL et d'autres sources. Hierarchical Clustering Explorer est un travail de recherche qui se focalise sur l'interface utilisateur, la visualisation et l'interaction [Seo and Shneiderman 2002]. Nous avons par ailleurs déjà mentionné des projets réutilisant les outils d'intégration de données comme GenoStar, Isymod, etc. Il en existe de nombreux autres.
- Les **approches langagières** permettent de concevoir à un haut niveau des petits programmes sous forme de scripts.
- Certaines **approches graphiques** (basées sur les flux de travail notamment) permettent de spécifier graphiquement des programmes équivalents.

Langages et scripts

On peut mentionner différents travaux : Catherine Letondal s'intéresse à la conception d'un langage de script adapté aux problématiques bioinformatiques et au biologiste peut familier de la programmation [Letondal 2001]. Des initiatives de plus grande ampleur sont déjà présentes : Matlab est généraliste, orienté calcul matriciel. SciLab, développé par l'INRIA en est un clone libre. Tous deux peuvent s'interfacer avec LabView (commercial) qui permet de générer des interfaces utilisateurs et des visualisations performantes. R est de la même façon un langage (libre) orienté statistiques et probabilités [Gentleman, Huber et al. 2005]. Il est particulièrement utilisé par les biologistes qui recourent depuis longtemps aux biostatistiques (et biostatisticiens). Il est fréquemment utilisé et a donné lieu à une initiative du nom de Bioconductor [Gentleman, Huber et al. 2005] : ce portail centralise de nombreuses bibliothèques, boîtes à outils et autres ressources pour l'utilisation de R en biologie. Du point de vue de l'intégration, des API permettent d'accéder à des SGBD en utilisant des requêtes SQL. D'autres API permettent d'interroger des portails en ligne : une API permet d'accéder à Entrez via les E-utilities du NCBI, à KEGG via SOAP, à BioMart (Ensembl), etc. De plus, de nombreux outils de visualisation sont aussi disponibles.

Services Web et flux de travail : vers la spécification graphique

> *« Les services Web sont une nouvelle race d'applications web. Ce sont des applications auto-descriptives, modulaires et faiblement couplées pouvant être publiées, localisées et invoquées à travers le web. Les services Web effectuent des tâches allant de simples requêtes à des processus métier plus complexes. » (IBM)*

Des plateformes permettent de spécifier des enchaînements d'invocations de services et proposer des outils pour articuler ces enchaînements (structure itératives, conditionnelles, etc.). Deux projets sont renommés en bioinformatique : myGrid et BioMoby. myGrid réunit notamment des acteurs du projet TAMBIS [Stevens, Robinson et al. 2003]. Le service y est décrit en utilisant DAML+OIL, un langage d'ontologie dont OWL est issu. L'objectif recherché est lié à la mise en œuvre d'une grille de calcul : parallélisation des services, performances, etc. BioMoby est un projet spécialisé dans les sciences du vivant qui se focalise sur l'intégration de données hétérogènes et distribuées [Kawas, Senger et al. 2006]. Il est aujourd'hui scindé en deux: Moby-S (Moby services) et S-Moby (Semantic Moby). Si les représentations diffèrent entre ces projets, la vision du point de vue de l'utilisateur reste similaire : des processus sont connectés par leurs entrées et leurs sorties. La figure 2.20 montre la description de deux services.

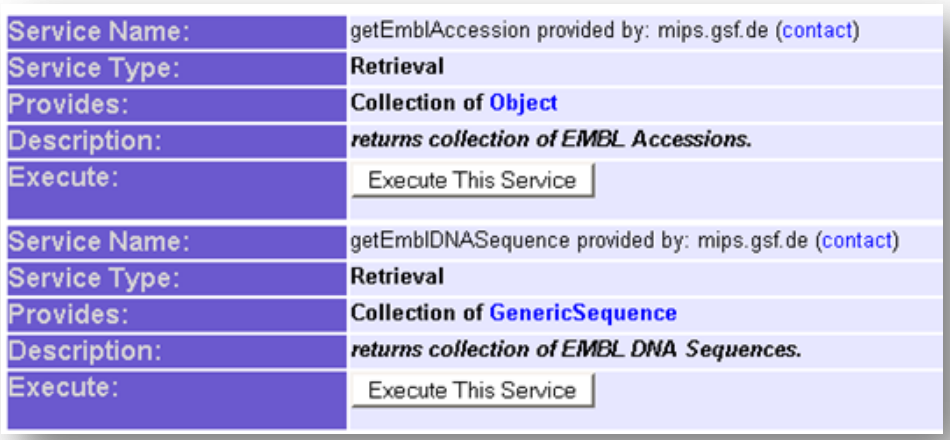


Figure 2.20 – Exemple de services décrits dans BioMoby.

Il est possible d'exploiter ces services en utilisant des API et des langages de programmation. Cependant, l'accès fréquent au réseau pour des tâches peu coûteuses devient rapidement une limite pour un usage intensif. L'intérêt réside dans le fait que de nombreux services sont autodécrits et hébergés à distance. Leur description est centralisée au sein d'annuaires. Il est ainsi possible d'utiliser des environnements graphiques pour exploiter ces services. On parle alors de *« flux de travail »* ou de *« flux de données »*[1]. Le plus fréquemment cité de ces environnements est Taverna [Hull, Wolstencroft et al. 2006]. Initialement construit pour myGrid, un plugin permet aussi d'utiliser BioMoby. Il est possible de spécifier un petit programme sous forme d'un graphe. L'atelier est illustré dans la figure 2.21. Il repose en particulier sur le langage de représentation de flux SCUFL (Simple Conceptual Unified Flow Language) représentable dans un format XML (XScufle).

[1] *« Workflow »* ou *« dataflow »*

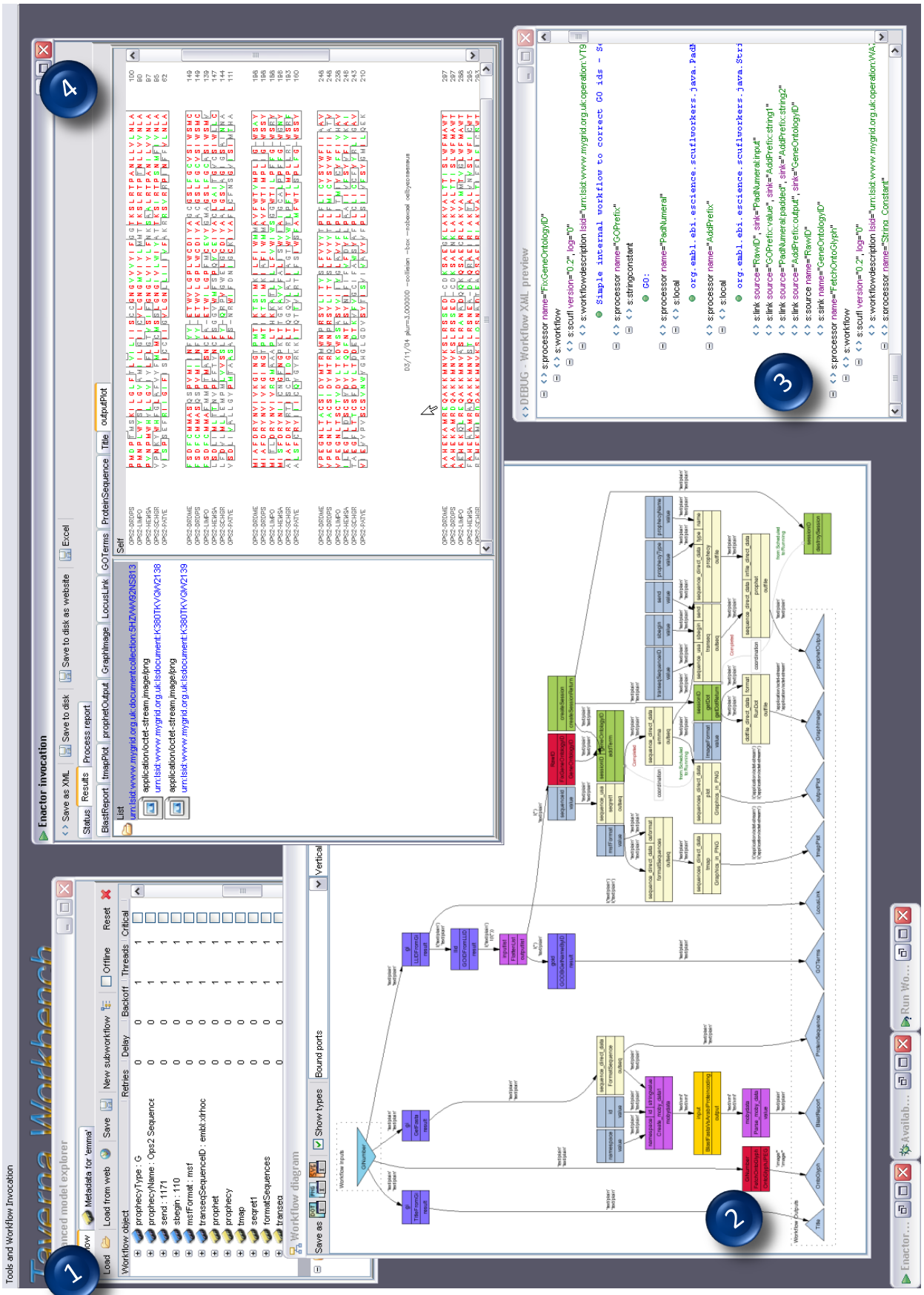


Figure 2.21 – Taverna

Légende figure 2.21 (page précédente) – Taverna

(1) modèles (types, etc.)
(2) spécification d'un enchaînement d'actions (sauvegardable)
(3) traduction sous forme de script
(4) résultat d'exécution

Les services permettent d'acquérir des données, d'appliquer des opérations (processus métiers, structures de contrôles conditionnelles et itératives, etc.) et de sortir un résultat sous forme graphique (ici un multi-alignement de séquence coloré).

Il existe d'autres environnements dédiés aux flux de travail : AGMIAL, Amadea, SEEK, DiscoveryNet, MHOLline, AdaptFlow, G-Pipe, BioPipe, ViPEr, PiPeLine Pilot, Vibe WildFire, Ptolemy II, Kepler, etc.

2.4 Synthèse

Ce chapitre a présenté deux domaines informatiques : la représentation des connaissances, et l'intégration des données. Ces deux problématiques sont liées, puisque les ressources terminologiques et ontologiques (RTO) sont un support sémantique aux données biologiques contenues dans les schémas, dans les annotations, c'est-à-dire dans les données et les métadonnées. Ces deux domaines, en dehors de leurs liens fonctionnels, ont des points communs : les RTO comme les systèmes d'informations sont nombreuses, hétérogènes. Elles sont décrites et structurées plus ou moins fortement et formellement, en fonction d'un besoin et d'un contexte. Les RTO et les systèmes d'information font partie du quotidien du biologiste et leur diversité est problématique pour cet utilisateur final, comme pour l'informaticien en charge de développer un outil.

Différentes solutions sont proposées pour pallier ces problèmes : des entrepôts comme UMLS intègrent des ontologies, des outils permettent leur alignement automatique. De même, de nombreuses solutions se sont intéressées à l'intégration de données issues de systèmes d'information divers. La technique ne peut pas, à elle seule, apporter une solution à ce problème qui est sociologique. Les solutions qu'elle apporte actuellement se destinent à différents usages :

- vue homogène d'un SGBD (système d'intégration) et standards d'interopérabilité à destination du développeur,
- portails, *workflows* et systèmes à base de chemins pour un besoin fonctionnel plus proche de l'utilisateur final.

Nous avons proposé une taxonomie de ces outils principalement basée sur la littérature existante. Le constat d'une grande diversité et hétérogénéité des approches est flagrant. Dans le chapitre suivant, nous proposons de revoir ces systèmes d'intégration suivant le point de vue de l'utilisateur. Sans revenir sur le constat de L. Stein, nous montrons comment l'informatique doit proposer un support technique adapté à la dimension sociologique du problème.

CHAPITRE 3

De l'intégration à la cartographie

« The challenges posed by drug discovery can be solved only if we can integrate data across many fields of life sciences »

TIM BERNERS-LEE

3.1 Introduction

La problématique, telle que nous l'avons présentée parait simple dans son expression. Elle peut se résumer dans une phrase fréquemment mentionnée par les utilisateurs biologistes :

J'ai souvent une vingtaine de fenêtres ouvertes sur mon bureau : je m'y perds

Cependant, derrière ce constat, c'est un problème large et complexe qui se pose. Sa solution nécessite l'implication de plusieurs communautés scientifiques. Les deux principales sont l'intégration de données hétérogènes et la conception d'interfaces hommes machines adaptées. Mais d'autres domaines peuvent être ponctuellement sollicités : analyse et fouille de données, représentation des connaissances, traitement automatique de la langue, apprentissage, etc.

Dans le chapitre précédent, nous avons détaillé les différentes approches de l'intégration de données d'un point de vue technique informatique. Nous abordons ici cette problématique du point de vue de l'utilisateur final : le biologiste. Nous montrons au travers d'expériences actuelles et concrètes que la dispersion de l'information subsiste et affecte le biologiste dans son quotidien.

Lincoln Stein évoque l'aspect sociologique du problème. Nous relativisions l'importance de ce problème dans la deuxième partie du chapitre en considérant que la technique n'a pas suffisamment pris en compte cet aspect. Parmi l'offre des systèmes existants, utilisateurs et développeurs diffèrent par leur besoins. Nous sommes convaincus qu'une solution pérenne doit proposer une réponse commune à leurs attentes.

Pour cela, nous proposons une approche extensible et visuelle de l'intégration : la carte. Elle est un outil pour se repérer dans un espace, choisir une direction et explorer cet espace, et enfin pour justifier, mémoriser, partager et capitaliser des connaissances. A l'image de la carte géographique, la carte de connaissance est spécifique à un besoin et un domaine. En revanche, la construction des cartes doit se faire au travers d'un cadre commun. La fin de ce chapitre dresse un état de l'art de la cartographie des connaissances, avec un intérêt particulier pour des connaissances biologiques.

3.2 Hétérogénéité et dispersion : un constat actualisé

Dans le chapitre précédent, nous avons réalisé un état de l'art taxonomique et technique de l'intégration de données. Dans cette section, nous adoptons le point de vue de l'utilisateur final et dressons un bilan des progrès pour cet utilisateur : le biologiste. Les approches détaillées dans le chapitre précédent sont synthétisées en adoptant son point de vue et nous montrons que deux grandes directions s'opposent : certaines approches répondent aux attentes du développeur, d'autres visent à proposer à l'utilisateur final un outil fonctionnel adapté.

Nous montrons au travers de deux expériences simples que malgré les nombreux progrès techniques réalisés en matière d'intégration, un problème subsiste pour l'utilisateur : les sources de données sont nombreuses, l'information est dispersées et complexe à analyser.

Cette section se termine par une discussion des aspects sociologiques et techniques du problème introduits par L. Stein [Stein 2003] et les controverses des contributions du domaine.

3.2.1 Synthèse des différentes approches de l'intégration du point de vue du biologiste

Approches formelles

Les **systèmes d'intégration** permettent une vue unifiée sur plusieurs bases de données à l'aide d'un langage de requête. Ce langage constitue leur principale force mais aussi leur principale limite, trop complexe pour un « non-informaticien ». Les requêtes permettent des

réponses multiples et précises, elles retournent un ensemble de tuples ou d'objets. L'entrepôt matérialise les données, principalement pour des raisons de performances et de propriété des données. Cette approche reste lourde : il faut dédier une machine, configurer et administrer un serveur, exécuter les procédures de mises à jour récurrentes qui nécessitent une bande passante importante, etc. Le médiateur offre l'avantage d'être bien plus simple à implanter et de consulter des données toujours à jour. Cependant il possède des limites majeures :

- l'utilisateur n'est pas propriétaire des données,
- les performances sont fortement réduites,
- la disponibilité du système dépend de la disponibilité des sources,
- les sources accessibles par un langage de requête sont peu nombreuses,
- les jointures entre différentes sources et l'intégration verticale sont difficiles à mettre en œuvre.

Les **langages de programmation** et de scripts sont puissants. Ils permettent de construire de véritables applications. Généralement, il n'y a pas réellement d'intégration. Le programmeur dispose des API ou services Web permettant d'accéder à certaines ressources. Les services Web reposent sur un ensemble de standards et proposent un accès alternatif au programmeur. Des outils de **workflow** permettent de spécifier des petits programmes graphiquement. Cela nécessite tout de même une rigueur formelle et un effort algorithmique au même titre que la programmation. Les outils de visualisation sont peu nombreux.

Toutes ces approches sont complémentaires et adressent des problématiques spécifiques. Dans la pratique, ils ne contribuent pas réellement à l'unification de la vue des données pour l'utilisateur. Ce sont des outils qui dépendent des contraintes du programmeur.

Systèmes orientés vers l'utilisateur

Les **navigateurs** reposent sur la nature du Web et proposent à l'utilisateur d'explorer l'espace informationnel en navigant au travers de liens. Les **systèmes à base de liens** les plus connus sont les principaux portails généralistes : LinkDB/DBGet, SRS, Entrez, etc. Pour l'utilisateur, aucune mise en œuvre n'est nécessaire, ils répondent à un besoin d'accès simple et généraliste à des banques de données. Les **systèmes à base de chemins** permettent de spécifier des requêtes en termes de chemins (scénarios). Les nœuds sont des pages et les liens sont des références entre ces pages. On peut comparer ces liens à la notion de jointure des SGBD, mais les requêtes sont plus intuitives pour l'utilisateur et peuvent parfois être spécifiées graphiquement. L'intégration y est lâche : SRS référence des documents, BioGuide et BioNavigation ne représentent que des types et des sources. Les navigateurs proposent une réponse fonctionnelle et simple pour l'utilisateur final, le biologiste. Leur principale lacune repose sur le manque de finesse de représentation, et le manque de richesse d'expression des requêtes. Les **plateformes** sont des logiciels dédiés à des tâches plus spécifiques et sont orientées vers l'utilisateur. Elles sont généralement fonctionnelles, mais l'intégration y est le plus souvent ponctuelle, faiblement réutilisable et extensible.

Bilan

Cette synthèse montre que les systèmes actuels divergent dans deux directions : une partie se focalise sur les attentes de l'utilisateur, l'autre vise à répondre aux besoins du développeur mais sont faiblement réutilisables et extensibles. Jusqu'ici, les approches n'ont pas proposé de réponse à ces deux acteurs. Peu d'outils mettent en œuvre des techniques de visualisation adaptées à la taille et à la nature des données. Et les portails et navigateurs qui sont les plus fréquemment utilisés ont une description *« lâche »* des données.

3.2.2 Hétérogénéité des interfaces

Jusqu'ici, nous avons abordé l'hétérogénéité suivant le point de vue d'un technicien, en la qualifiant de structurelle, verticale, horizontale, etc. De nombreuses contributions ont permis de

pallier ce problème ; ce n'est pourtant pas aujourd'hui le ressenti du biologiste. Lincoln Stein a montré qu'en effet, l'hétérogénéité commence dès la première page pour l'utilisateur, au sein de l'interface [Stein 2003]. Il a proposé une expérience en comparant trois outils similaires, Ensembl, FlyBase et UCSC Genome Browser. Nous avons reproduit cette expérience quatre ans plus tard sur trois portails centrés sur la génomique : Entrez Gene, GeneDB et PlasmoDB (GUS).

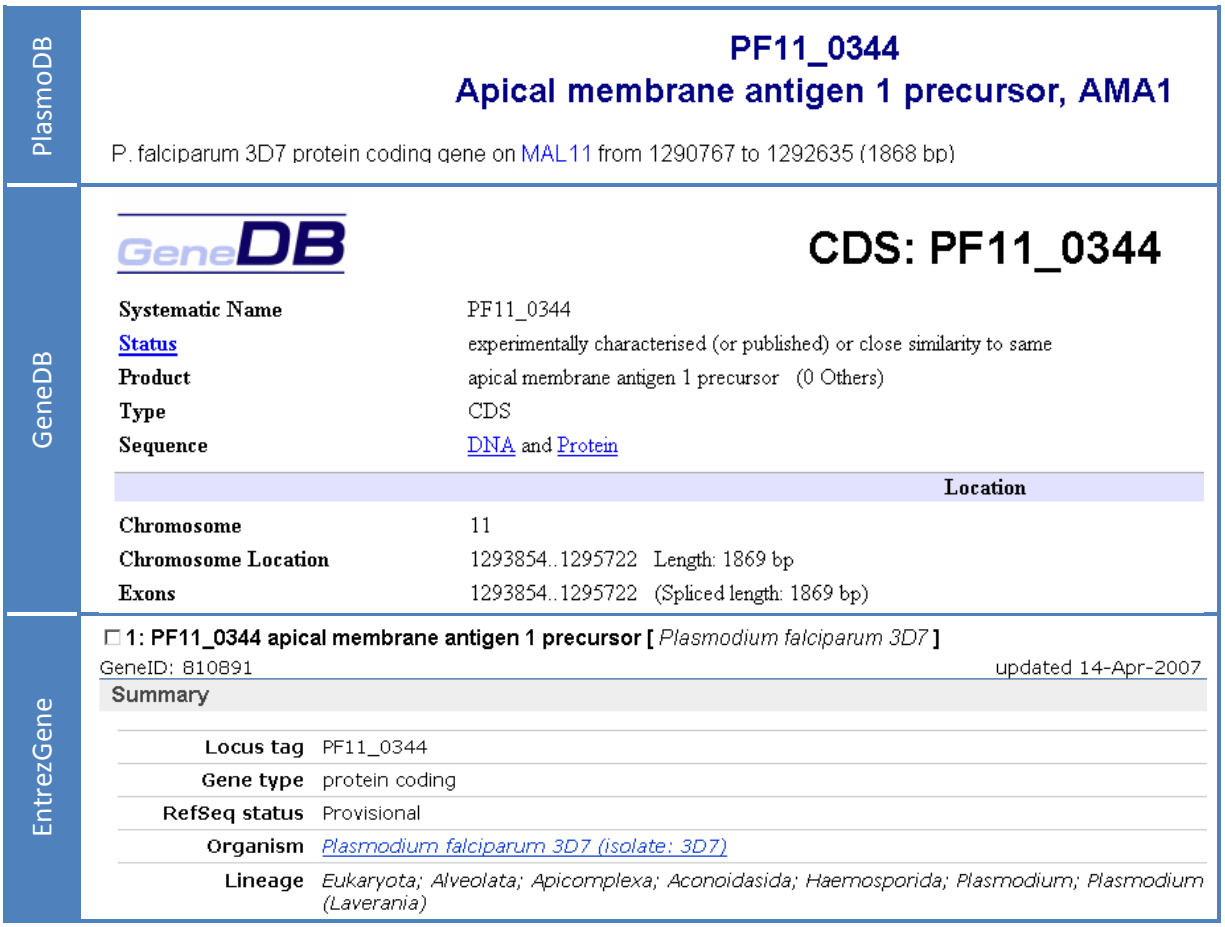


Figure 3.1 – Comparaison de l'information résumée pour le gène PF11_344 dans trois portails poursuivant des objectifs voisins et comportant des données similaires.

Par soucis de concision, nous nous focalisons sur le premier écran de chaque page, mais nous recommandons au lecteur de prolonger cette expérience. Notons que GeneDB est officiellement une source de PlasmoDB. Le gène étudié est le même (PF11_0344). On retrouve, sur le premier écran de chacun, une information globalement homogène : un titre, un résumé, et un navigateur pour consulter le génome. Si on s'intéresse plus en détail aux informations présentes sur les premières pages, on peut cependant constater de nombreuses divergences (figure 3.1) :

- Le titre diffère : GeneDB indique l'identifiant et l'acronyme CDS qui indique qu'il s'agit d'une séquence codante. Entrez Gene et PlasmoDB décrivent une annotation du gène (ce qui induit qu'il est supposé codant).
- Les noms de champs possèdent parfois la même information mais ont un ordre et des intitulés différents.
- La séquence possède (presque) la même longueur, mais dans GeneDB, la localisation est décalée.
- GeneDB est le seul à stipuler comment l'annotation a été obtenue.
- GeneDB est le seul à préciser le chromosome sur lequel est situé le gène, mais l'utilisateur averti sait que la dénomination *« PF11_* »* ou *« MAL11* »* indique qu'il s'agit du chromosome 11 de *Plasmodium Falciparum* (ou *Malaria*).

La seconde composante principale du premier écran est le navigateur interactif de génome (*« genome browser »*). Il faut noter que GeneDB et PlasmoDB utilisent initialement le même outil pour explorer le génome. On constate de nouvelles divergences (figure 3.2) :

- Le code de couleurs n'est pas le même entre PlasmoDB et GeneDB, et aucune documentation n'en indique la signification. Entrez Gene ne propose que deux couleurs pour indiquer s'il s'agit d'une séquence codante ou non.
- L'échelle varie dans les trois vues. L'amplitude pour PlasmoDB est de 14 000 paires de bases, près du double pour GeneDB, alors qu'Entrez Gene se limite à la séquence recherchée (1869 paires de bases). PlasmoDB propose une graduation sur l'échelle plus précise : ce choix peut se justifier par la largeur de portion du chromosome visualisée.
- Entrez Gene indique l'orientation de la séquence (5'-3') sur l'échelle, PlasmoDB permet de voir l'orientation de la séquence en la symbolisant par une flèche.
- PlasmoDB permet de visualiser une annotation sommaire pour chaque gène représenté.
- Dans PlasmoDB et Entrez Gene, le chevron vertical présent dans un gène indique qu'il s'agit d'un gène nettoyé (*« curated »*).

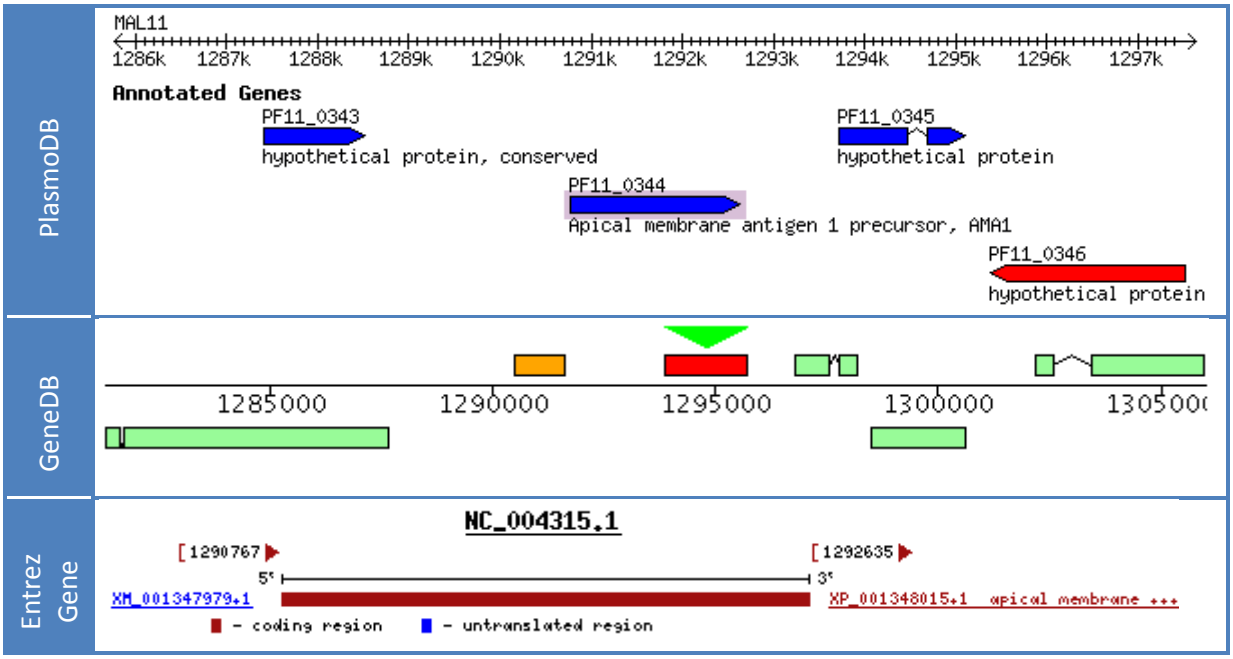


Figure 3.2 – Composante principale des interfaces « Genome Browser » proposées par les portails PlasmoDB, GeneDB et Entrez Gene.

D'un point de vue interactif, là encore, les interfaces diffèrent (figure 3.3). Entrez Gene n'affiche aucune information pertinente au survol du navigateur. Par contre, si l'on clique sur un élément, un menu contextuel permet de suivre un lien vers d'autres systèmes d'information du portail Entrez (Nucleotide/GenBank, Protein, Blink et CDD). GeneDB propose une information très sommaire : le nom du gène et une annotation. Ces deux informations sont présentes dans l'affichage permanent de fond du navigateur de PlasmoDB. Ce dernier propose un menu contextuel plus esthétique et complet indiquant la lignée cellulaire et l'espèce concernée, le type de gène, et sa localisation. Enfin, il propose un lien pour télécharger la séquence nucléotidique codante et la séquence protéique au format Fasta.

Ces portails sont assez intuitifs dans leurs fonctions élémentaires. Ils poursuivent les mêmes objectifs et contiennent une information similaire. On constate cependant des différences alors même que ces sources sont synchronisées avec les sources de référence. Le biologiste qui souhaite une information fiable et complète doit se référer à plusieurs systèmes et comparer leurs contenus. Les divergences entre les interfaces rendent cette tâche longue et difficile. De

plus, lorsque les données divergent, il peut être impossible de savoir objectivement quelle source est la plus fiable. Dans la pratique, on constate plus des phénomènes de croyances de confiances liées à une réputation de la source. La découverte d'un système d'information et son estimation sont alors assez aléatoires et subjectives, basées sur des expériences personnelles, le ouïe-dire, l'esthétique et la facilité d'utilisation, la rapidité de réponse ou la disponibilité, la réputation de l'entité dirigeante (EBI, NCBI, …), etc.

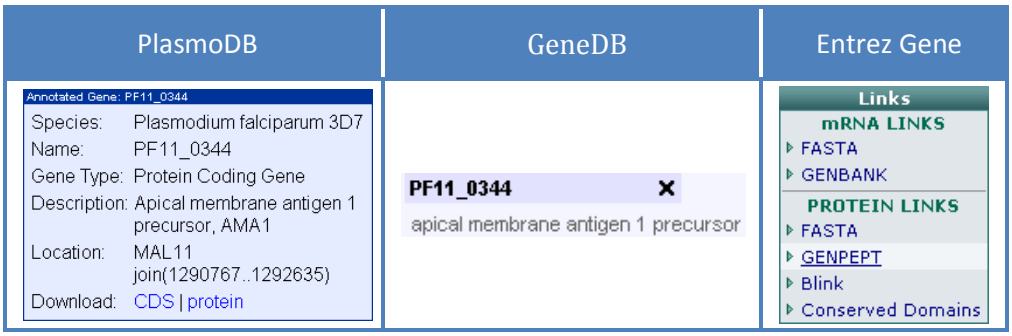


Figure 3.3 – Menus contextuels des interfaces « Genome Browser » proposées par les portails PlasmoDB, GeneDB et Entrez Gene.

Lorsque l'utilisateur souhaite obtenir une information complète sur un gène, il est contraint de comparer les informations de plusieurs applications. Il manipule donc un nombre raisonnable de fenêtres (par exemple 2 à 5). Cette tâche est rendue plus difficile quand l'information ne suit pas le même ordre et la même présentation. Lorsqu'il souhaite analyser les données d'un ensemble plus important de gènes, des dizaines de fenêtres sont ouvertes et l'utilisateur ne peut traiter simultanément une telle quantité d'information. Une pratique courante consiste à utiliser un tableur comme structure de données intermédiaire. Comme nous posons une addition ou une multiplication avec un papier et un crayon, le tableur permet de structurer quelques éléments d'information importants. On parle *d'amplification de la cognition externe*. Bien sûr cette tâche nécessite du temps et s'avère parfois fastidieuse. Mais le plus troublant est qu'initialement, le biologiste nous décrivait les problèmes qu'il rencontrait pour croiser des informations. Ce que réalise le biologiste au travers de ce tableur est l'une des principales tâches que devrait réaliser le système d'intégration pour lui.

3.2.3 Un réseau de sources complexe

La nature même du Web repose sur une navigation entre des pages au travers d'hyperliens. Il est difficile pour le chercheur de se tenir informé des ressources existantes, correspondant à ses besoins, et d'obtenir une information sur la qualité et la fiabilité de ses données. Le plus souvent, le biologiste découvre les ressources au fil des liens parcourus, de façon assez aléatoire. Pour accéder à la source, il mémorise parfois le chemin qui l'y a conduit et non l'adresse de la source.

Pour illustrer la complexité du réseau de sources, nous avons choisi de rechercher un gène sur Entrez Gene puis de parcourir toutes les sources accessibles à une distance de deux clics. Notre expérience nous a permis de construire le graphe dessiné dans la figure 3.4. Ce graphe n'est qu'une version simplifiée de la réalité ; certains aspects n'apparaissent pas dans ce graphe :

- les références issues de sources à une distance 2 d'Entrez Gene,
- les références internes,
- les outils de recherche, les différents services, les options de paramétrage ou de téléchargement,
- la multiplicité des références ; une page d'une source référence parfois jusqu'à plusieurs dizaines de pages d'une autre source.

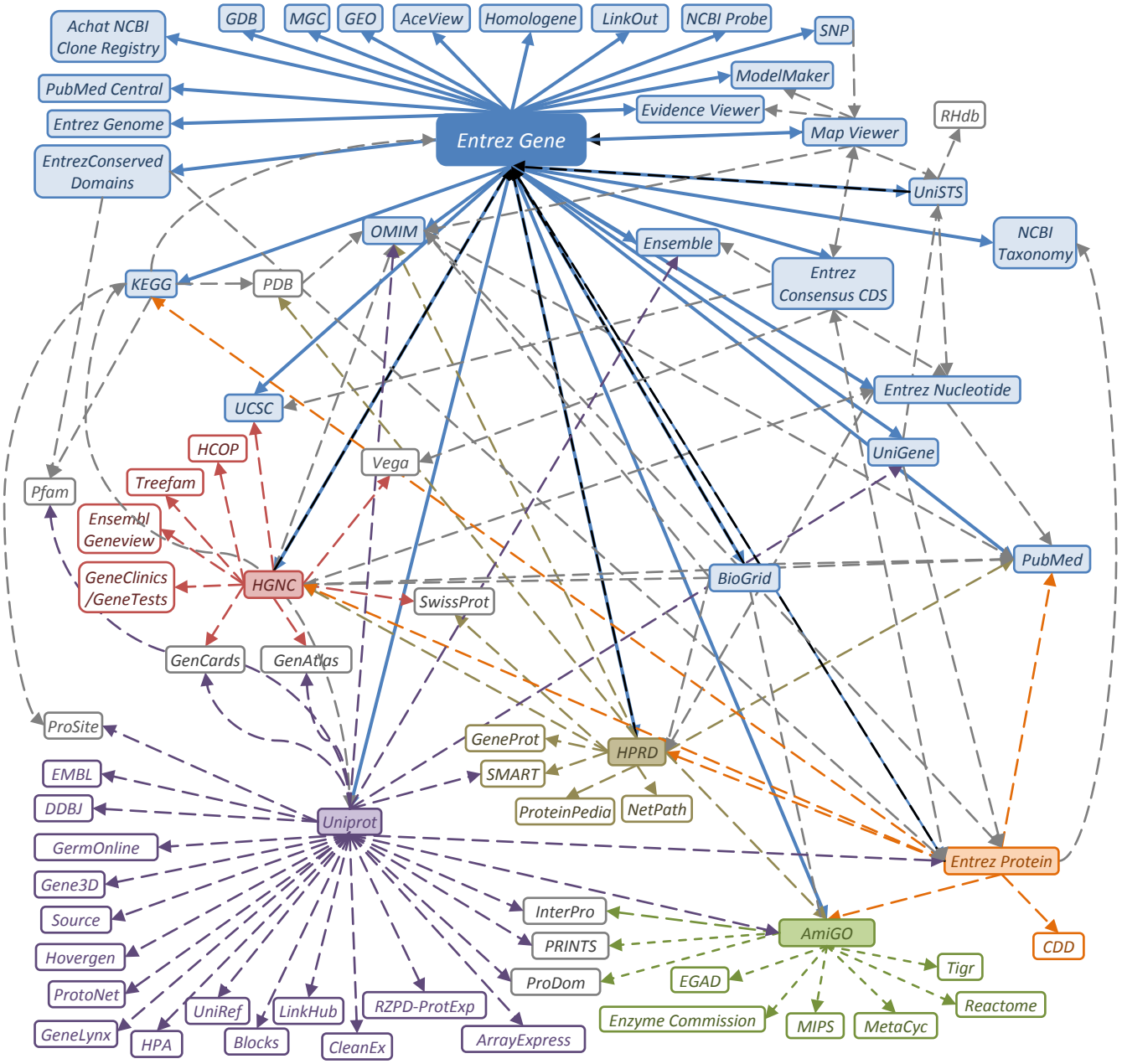


Figure 3.4 – Liste des sources accessibles en deux clics à partir d'Entrez Gene. Les boîtes pleines sont reliées par une flèche continue à Entrez Gene ; il s'agit des éléments accessibles à une distance de 1 clic. Les boîtes sur fond blanc et les flèches discontinues correspondent aux autres références croisées. Nous n'avons pas représenté les références croisées entre ces sources qui correspondraient à un troisième clic. Les couleurs ont été utilisées pour mieux suivre les origines et groupes de dépendances.

En définitive, des centaines de chemins sont proposés à l'utilisateur. Ce problème est présent au niveau de l'ensemble des sources, et on pourrait penser qu'il disparait lorsque l'utilisateur se restreint à un seul portail. Pourtant, il persiste. SRS référence actuellement 400 sources différentes. Bien que hiérarchisé, ce registre de bases de données n'est pas maitrisé par l'utilisateur. Plus éloquent, Entrez, qui ne possède que 25 sources, éprouve le besoin de schématiser et documenter les croisements entre des bases de données (figure 3.5). La copie d'écran témoigne, à nouveau, de la difficulté à interpréter la structure de l'espace informationnel.

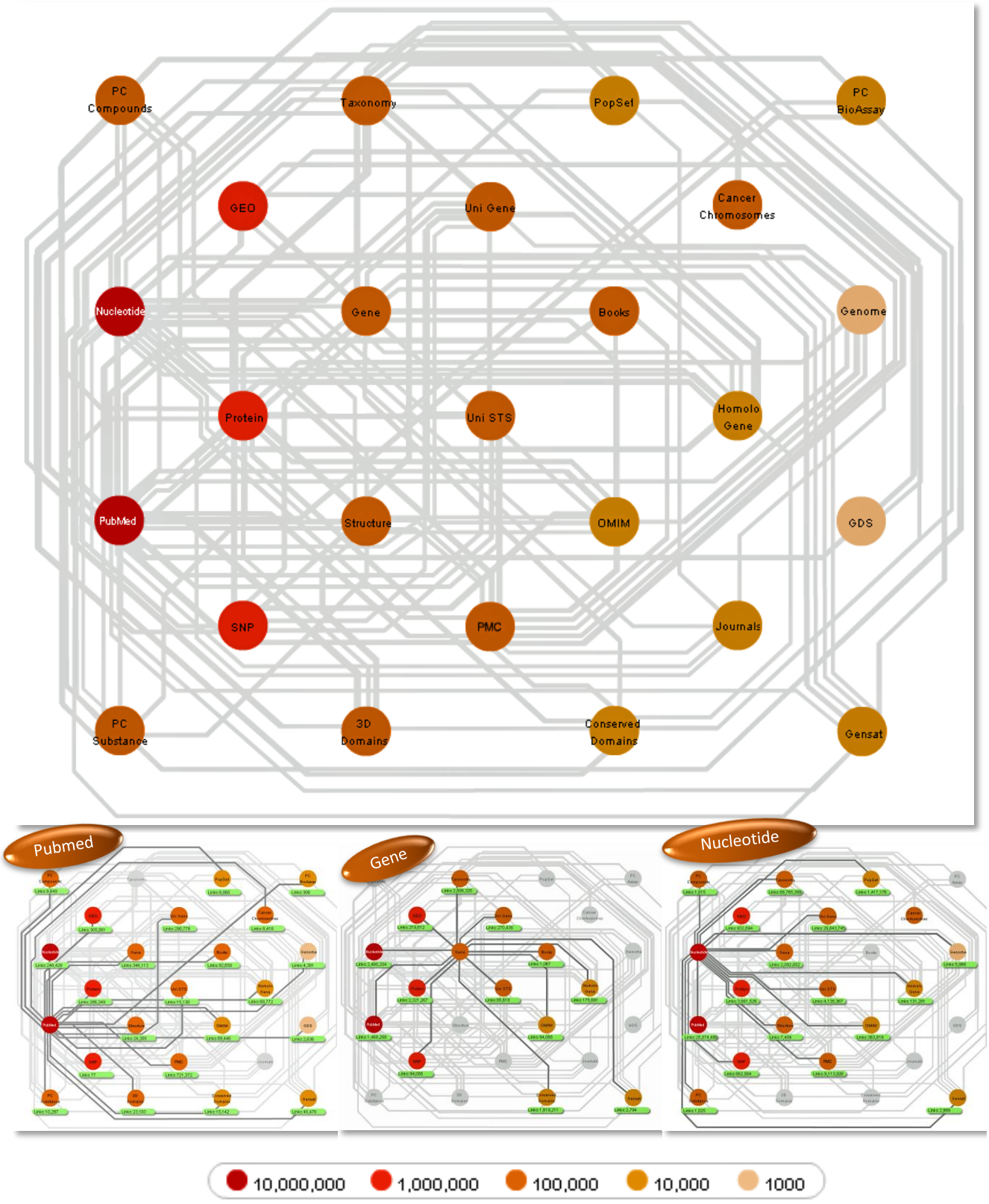


Figure 3.5 – Schéma d'interconnexion dans le portail Entrez. A droite, les figures réduites mettent en évidence les références croisées vers d'autres sources : en haut Nucleotide, au milieu Gene, en bas PubMed. La couleur des sources correspond à l'ordre de grandeur du nombre d'enregistrements contenus dans chaque source. Cela met en évidence au sein d'un même portail la complexité des références croisées existantes et la difficulté pour l'utilisateur de se repérer dans ce réseau de ressources.

3.3 Bilan et directions pour améliorer le quotidien du biologiste

3.3.1 Bilan suivant différents points de vue

Une avancée pour l'utilisateur final ?

De nos discussions avec les biologistes est ressortie une problématique de façon récurrente : ils sont perdus sur leur espace de travail, au milieu de l'information, et passent un temps important à croiser cette information et la synthétiser dans leur tableur favori.

Lorsqu'ils disent qu'ils sont perdus, ils n'expriment pas simplement une surcharge de leur bureau sur l'écran, encombré par un grand nombre de fenêtres. Ceci n'est qu'un symptôme et une mesure du problème, mais le problème ne se restreint pas à cela. L'utilisateur est perdu parce qu'il ne connait qu'une faible partie des sources, ne peut pas estimer leur qualité, n'a pas de moyen de contrôle, et le plus souvent ne peut pas accéder à une traçabilité suffisante. Lorsque deux annotations diffèrent, il n'a pas de moyen de savoir si l'une des deux est obsolète, et laquelle. L'utilisateur est perdu car il ne maitrise pas l'inventaire des sources, leur contenu, leur mise en forme. A tous ces niveaux, la quantité et l'hétérogénéité l'empêchent d'avoir une représentation en mémoire de cet ensemble de sources. Il se résout donc à utiliser quelques portails de référence, qui sont choisis le plus souvent suivant des critères aléatoires et subjectifs.

Concernant la problématique concrète de l'utilisateur, nous avons vu que l'intégration de données ne propose à l'heure actuelle que des solutions partielles et ponctuelles. Quand bien même le besoin d'un utilisateur est rempli par un système, il possède généralement d'autres besoins auxquels il ne trouve aucune réponse efficace. Nous proposons quatre questions auxquelles les systèmes d'intégration doivent répondre pour satisfaire les besoins de l'utilisateur final :

- Combien de fenêtres l'utilisateur manipule-t-il ?
- Combien d'environnements[1] distincts consulte-t-il régulièrement et directement ?
- Utilise-t-il encore des copier/coller pour rassembler l'information au sein de son tableur favori ?
- L'utilisateur doit-il avoir une connaissance et compréhension de l'inventaire des sources ?[2]

Ces questions sont simples. Elles permettent dès le début d'un projet de prévoir objectivement si la solution envisagée améliorera le quotidien du biologiste. A la fin du projet, elles permettent d'évaluer le bénéfice produit. Lorsque l'on prend du recul sur les approches précédemment décrites et qu'on les confronte à ces questions, on comprend rapidement pourquoi l'utilisateur final n'a pas ressenti d'amélioration significative dans son quotidien.

Un constat général sur lequel nous terminons ce bilan est qu'à chaque besoin est proposée une approche spécifique. L'utilisateur possédant de multiples besoins, il est amené à faire cohabiter plusieurs approches sur son espace de travail. Pour unifier la perception de l'information à l'utilisateur, il est donc avant tout nécessaire d'unifier les approches de l'intégration de données.

[1] Par environnement, nous considérons ici les sites, portails, logiciels et applications qu'il manipule.

[2] Cette question est déjà présente dans [Cohen-Boulakia 2005]

Sociologique ou technique ?

> *The IGD project survived for slightly longer than a year before collapsing. The main reason for its collapse, as described by the principal investigator on the project (O. Ritter, personal communication), was the database churn issue. On average, each of the source databases changed its data model twice a year. This meant that the IGD data import system broke down every two weeks and the dumping and transformation programs had to be rewritten — a task that eventually became unmanageable. […]* [1]
>
> *Although it is tempting to treat the integration of biological databases as a technological problem, in fact the main impediment to achieving this goal is not technological but sociological. In the opinion of this author, meaningful scaleable integration cannot be achieved without the cooperation of the data providers. As long as the data providers continue to produce online databases without regard for the way in which the information will be aggregated, integration will be a monumental task. However, in the absence of accepted standards for the representation and exchange of biological data, it is far from simple for data providers to achieve the goal of making their data available in a form that can be cleanly integrated and maintained.* [2] ***Lincoln Stein [Stein 2003]***

Nous partageons l'opinion de Lincoln Stein et réalisons les mêmes constats [Stein 2003]. Le problème est sociologique plus que technique, les producteurs de données ne doivent pas « développer dans leur coin ». En témoigne la diversité d'accès aux données (fichiers, services, …), de formats (GenBank, Fasta, XML,…), et des langages de requête, la disponibilité de mises à jour incrémentales ou d'historiques, et le défaut d'utilisation de certains standards (LSID, …). L. Stein indique que la technique n'a cependant pas fourni les outils suffisants pour la représentation et l'échange des données. Aujourd'hui, de nombreux standards adaptés ont été proposés et ne sont pas utilisés pour autant. Cela s'explique ; la conception d'un entrepôt provient souvent d'un besoin de propriété de données et d'un système, de besoins nouveaux. La recherche est telle qu'il y a donc une volonté d'autonomie, d'indépendance, de se distinguer dans la communauté, tout en possédant des budgets qui obligent à établir certaines priorités : les fonctions essentielles de l'utilisateur qui dirige le projet. A chaque problème nouveau une solution nouvelle.

Pour adresser ces problèmes, plusieurs sujets doivent être traités simultanément et dirigent les choix dans notre environnement :

- L'intégration doit être peu coûteuse, et mutualisée autant que possible, afin de pallier les difficultés décrites au sein du projet IGD. Pour cela, elle doit posséder un modèle simple.
- Elle doit prendre en compte les besoins de l'ensemble des développeurs et répondre à leurs contraintes : entrepôt vs légèreté, requête SQL vs moteur de recherche textuel, …

---

[1] *Le projet IGD a survécu pendant plus d'un an avant de s'effondrer. La principale raison de cet effondrement, décrite par le principal instigateur du projet (O. Ritter, communication personnelle), était le brassage des bases de données. En moyenne, chaque base de données sources changeait son modèle deux fois par ans. Ceci signifiait que le système d'importation des données tombait en panne toutes les deux semaines et que les procédures de téléchargement et de transformation des données devaient être réécrites - une tâche qui est devenue par la suite ingérable. […]*

[2] *Bien qu'il soit tentant de traiter l'intégration de bases de données biologiques comme un problème technologique, en fait le principal empêchement pour atteindre cet objectif n'est pas technologique mais sociologique. D'après l'auteur, une intégration significative à l'échelle du problème ne peut être réalisée sans la coopération des producteurs de données. Tant que ceux-ci continueront à produire des bases de données en ligne sans tenir compte des méthodes d'agrégation qui leur sont appliquées, l'intégration sera une tâche monumentale. Cependant, en l'absence de standards acceptés pour la représentation et l'échange de données biologiques, il est loin d'être simple pour le fournisseur des données d'atteindre l'objectif de rendre ses données disponibles sous une forme qui peut être proprement intégrable et maintenable.*

- Elle doit rendre transparentes et immédiates les fonctionnalités souvent mises de côté pour des questions de coût mais importantes : traçabilité, standards d'interopérabilité, extensibilité, etc.
- Elle doit prendre en compte les besoins des utilisateurs : proposer plusieurs solutions d'accès visuel (workflow, portails, etc.) et permettre d'accéder à l'information depuis les outils métiers. Elle doit envisager plusieurs outils métiers, et mettre à disposition des outils de visualisation polyvalents et correspondant aux besoins variés.
- L'environnement doit fédérer et attirer les décideurs et les développeurs : il doit les contraindre à être interopérables tout en réduisant le coût de leur projet, le temps de développement et en fournissant des outils pour l'utilisateur final.

Une réduction de l'hétérogénéité ?

Le témoignage apporté par le projet IGD met en évidence un constat : l'évolution rapide des sources met en échec les approches médiateur. L'entrepôt pallie ce problème : dans le cas d'un problème de connexion avec une source, le système est disponible, le problème relève de la mise à jour des données. L'entrepôt offre de plus la propriété physique des données, importante pour nettoyer les données ou pour des contraintes de performances et de confidentialité. Les entrepôts ont donc remporté un vif succès : par exemple GUS a été implanté plus de 20 fois, et AceDB plus de 50.

Cependant, en facilitant la mise en œuvre d'entrepôts, la communauté a favorisé leur multiplication. Grâce à l'intégration, certains utilisateurs disposent de sources plus adaptées. Mais plus globalement, la multiplication des sources ne fait qu'accroître le problème de l'hétérogénéité. L'intégration n'est jamais parfaite, et découle des pertes de traçabilité et de métadonnées, d'une augmentation de l'hétérogénéité et de la redondance. L'utilisateur qui veut croiser des données doit parcourir plus de sources. Il doit traiter plus d'information redondante, et plus d'information divergente. La redondance est une perte de temps, la divergence est un problème plus grave encore. Lorsqu'une donnée est mise à jour et répercutée sur une partie des sources, il n'a plus de moyen de savoir quelle est l'origine de la donnée, la version la plus à jour, la façon dont elle a été produite, la raison pour laquelle elle a été modifiée.

Une pratique courante du biologiste est d'utiliser des outils généralistes, et un portail communautaire. Lorsque les données divergent, il doit faire confiance au portail spécialisé, sans réelle preuve de la qualité et de la fiabilité des données. Le choix d'une ressource est le plus souvent assez subjectif et repose sur des critères d'esthétique et de simplicité d'utilisation de l'interface, de rapidité et de disponibilité de la source, de notoriété de l'institut propriétaire, ou sur la recommandation par des collègues, etc.

3.3.2 Directions choisies pour une réponse commune aux développeurs et utilisateurs finaux

Le bilan précédent soulève plusieurs problèmes qui nécessitent une solution commune. Rappelons que notre problématique concrète se résume simplement : Comment réduire le nombre de fenêtres ? Comment éviter à l'utilisateur de faire des copier/coller fastidieux dans son tableur favori pour accéder à une information synthétique ? Nous somme confrontés à deux utilisateurs qui ont un pouvoir de décision :

- l'utilisateur final, le biologiste, qui manipule l'outil et pour lequel on souhaite améliorer l'accès à l'information,
- le développeur qui conçoit et implante l'outil pour l'utilisateur final, qui est généralement autonome concernant les choix techniques.

Nous sommes convaincus que l'adoption de notre solution n'est réaliste que si elle répond simultanément aux *desideratas* des utilisateurs et des développeurs, en s'intéressant de façon transversale à plusieurs domaines : l'intégration de données et la visualisation d'information.

Pour atteindre cet objectif, il nous semble indispensable de concilier plusieurs directions dans nos recherches. Pour réduire le nombre de fenêtres manipulées par l'utilisateur, il faut accéder à l'information au sein des outils métiers de l'utilisateur. Cet utilisateur doit pouvoir visualiser et manipuler l'information synthétisée de façon adaptée à la tâche. Les pages Web des portails ont montré leurs limites. L'information manipulée par ces outils est la même, les tâches réalisées par l'utilisateur dans différents outils sont le plus souvent liées. Il faut donc que ces applications soient interopérables. La donnée doit être centralisable, partageable entre plusieurs utilisateurs et plusieurs applications. Enfin, si notre projet vise à proposer un nouveau point de départ pour des projets l'adoptant, de nombreux outils existent (entrepôts, médiateurs, plateformes, portails, etc.). La migration ne peut être que progressive, il est donc nécessaire que notre projet s'ouvre à toutes ces approches. Les outils existants doivent pouvoir s'étendre vers notre environnement et être interopérables facilement. La suite de cette section présente ces trois directions. Nous verrons dans le chapitre suivant comment nous les concilions.

Accès à l'information depuis l'outil métier

La figure 3.6 schématise l'évolution de l'intégration du point de vue de l'utilisateur : en introduisant des systèmes d'intégration et des portails (a→b), la communauté a permis de réduire le nombre de sources. Cependant, l'information n'est pas accessible directement dans l'application métier permettant l'analyse de données expérimentales. L'utilisateur est amené à consulter plusieurs portails et à ouvrir de nombreuses pages dans ces portails. Nous prônons d'intégrer l'information dans les outils métiers de l'utilisateur (b→c).

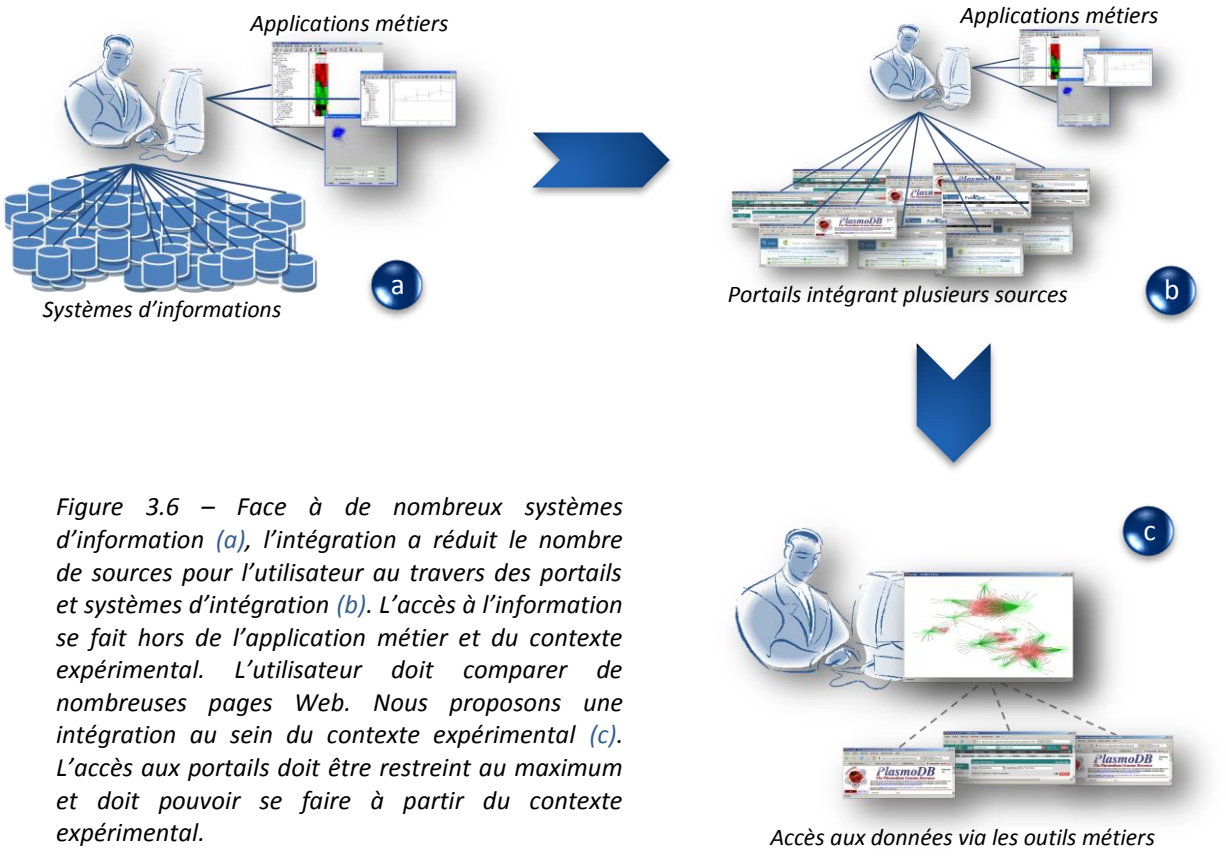


Figure 3.6 – Face à de nombreux systèmes d'information (a), l'intégration a réduit le nombre de sources pour l'utilisateur au travers des portails et systèmes d'intégration (b). L'accès à l'information se fait hors de l'application métier et du contexte expérimental. L'utilisateur doit comparer de nombreuses pages Web. Nous proposons une intégration au sein du contexte expérimental (c). L'accès aux portails doit être restreint au maximum et doit pouvoir se faire à partir du contexte expérimental.

Visualisation

L'environnement permet d'accéder à une grande quantité de données complexes. Par conséquent, il est nécessaire de mettre à disposition de l'utilisateur des mécanismes de visualisation et d'interaction adaptés en s'appuyant sur l'état de l'art. Ces outils de visualisation sont par ailleurs des services attractifs pour fédérer les développeurs. L'utilisateur manipulant plusieurs applications métier, la visualisation doit homogénéiser la manipulation des données

entre les différentes applications et être capable de s'adapter aux différentes tâches réalisables dans ces applications : les besoins d'un utilisateur d'une veille bibliographique ne sont pas les mêmes qu'une analyse de données d'expression. L'usage diffère quand on analyse 5 gènes, 50, ou 500. Il ne s'agit pas simplement de proposer un zoom optique, mais de considérer plusieurs cas d'utilisation.

L'informatique est récente dans le quotidien du biologiste. Ce dernier utilise toujours son traditionnel cahier d'expériences. Il est donc nécessaire de pouvoir figer l'information à un instant donné, la sauvegarder sous forme de fichier, d'image et de l'imprimer pour la griffonner et la coller sur le cahier d'expériences. Plus généralement, il est important de s'ouvrir, de respecter les pratiques du métier, et de prévoir l'existence de plusieurs supports.

La donnée au centre de l'interopérabilité

Plusieurs applications coexistent, et les utilisateurs travaillent en équipe le plus souvent. Il est donc nécessaire de permettre à différentes applications de partager les mêmes données, et à différents utilisateurs d'y accéder simultanément. Les tâches réalisées au sein d'un outil et un utilisateur doivent être répercutables immédiatement dans un autre outil manipulé par une autre personne. Cela implique une centralisation des données, la gestion d'une traçabilité et l'utilisation de standards d'interopérabilité. La donnée est l'objet mais aussi le pivot de cette interopérabilité (figure 3.7).

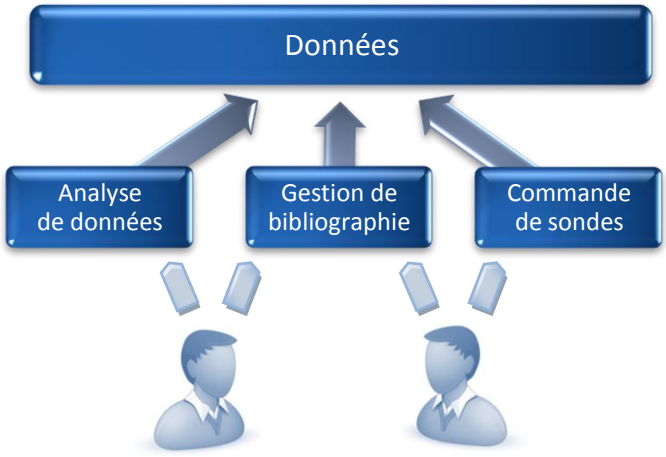


Figure 3.7 – La donnée est l'élément central sur lequel porte l'interopérabilité. Elle est partagée par les outils et les utilisateurs. Cette fonctionnalité ne doit pas représenter une charge supplémentaire pour le développeur, et doit être transparente pour l'utilisateur.

Socle commun et fédérateur

Notre objectif concerne à la fois l'utilisateur et le développeur. Notre hypothèse se base sur le constat de l'existence de différentes approches de l'intégration de données. Ces approches n'apportent que des réponses partielles, favorise la multiplicité d'outils tout en ne contribuant pas à leur interopérabilité. Si on souhaite homogénéiser le cadre applicatif de l'utilisateur, il faut auparavant homogénéiser celui du développeur et proposer une approche commune permettant de concilier les principaux avantages de différentes approches et pallier leurs limites. Il ne s'agit pas de les remplacer, mais de leur permettre de cohabiter (cf. figure 3.8).

Pour fédérer les développeurs, nous devons leur proposer un cadre commun pour des besoins variés. De plus, pour les attirer, nous devons leur fournir certains services (API métier, visualisation, gestion de formats de fichiers, etc.). Enfin, nous ne devons pas les repousser : les fonctionnalités supplémentaires que nous proposons doivent être accessibles sans effort pour lui (traçabilité, multiplicité des supports, etc.).

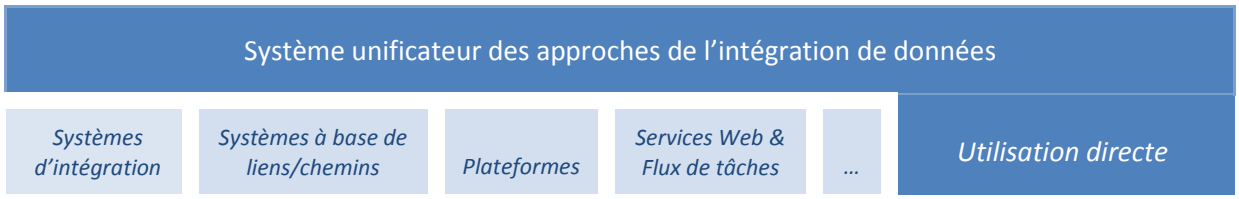


Figure 3.8 – Notre approche (blanc sur bleu foncé) se positionne comme une couche unificatrice favorisant l'interopérabilité des approches existantes. Elle propose un ensemble de services et d'outils fonctionnels permettant le développement rapide d'applications métiers accédant aux données intégrées et les visualisant.

3.4 Cartographie des connaissances, fondements et mises en œuvre.

L'adoption d'une métaphore est fréquente dans la conception d'un outil. Notre motivation n'est alors pas *« la beauté de la métaphore »*. Nous poursuivons l'objectif de réduire l'apprentissage de l'outil et d'en améliorer la perception et la compréhension. Dans la métaphore, l'image du référant est conditionnée par celle du référé. Les conséquences sont doubles. La première est à un niveau « marketing » : si l'image du référé est perçue comme positive, alors le référant peut bénéficier d'un *a priori* positif. Mais la conséquence la plus importante est la possibilité de transposer la sémantique du référé à celle du référant. En réutilisant la sémantique, on souhaite réduire le temps d'apprentissage de l'outil. Ceci est d'autant plus facile que l'utilisateur est habitué à manipuler des outils similaires.

Nous avons choisi la métaphore de la cartographie. La cartographie est la discipline, la technique ou l'activité de construction de cartes. Pour transposer les motivations présentées dans le paragraphe précédent, la carte représente généralement un objet simple, utile, nécessaire au quotidien. Elle évite de se perdre, permet de se positionner et de trouver son chemin. La carte permet d'adopter un nouveau point de vue sur le monde à différentes échelles et peut contenir des données abstraites. Elle revêt ainsi un intérêt explicatif, argumentatif et didactique. Elle permet de simplifier une information, facilite sa compréhension et sa mémorisation. Elle est utilisée pour faire émerger une connaissance, la partager, l'archiver ou la communiquer.

Cette caractéristique de la carte est déjà exploitée dans certains grands projets de la biologie. La cartographie du génome est un projet de séquençage de génomes et d'annotation des gènes qui sont identifiés et localisés. Les cartes des voies métaboliques synthétisent dans de grands graphes des mécanismes biochimiques de la cellule.

De plus, la carte possède une sémantique proche de certaines thématiques que nous avons mentionnées : systèmes à base de chemins, données multidimensionnelles et relationnelles, besoin de visualisation pour explorer un espace informationnel, etc.

Dans la dernière partie de ce chapitre, nous proposons un rapide état de l'art de la cartographie. Après un résumé historique, les fondements et justifications théoriques de nos choix sont présentés plus en détail. Nos motivations apparaissent alors avec plus de clarté. Par la suite, nous abordons la cartographie en énumérant les principales approches, nous les discutons à partir des principaux exemples que l'on recense dans la littérature, avant de préciser leur application dans le domaine biomédical.

◀*Figure 3.9 – Tablette d'argile, « carte babylonienne du monde ». Il s'agit de la plus ancienne carte connue. Elle remonterait à 2500 ans avant JC pour certains, en 500 et 700 ans avant JC selon d'autres sources.*

Figure 3.10 – La carte T-in-O, Orbis Terrarum, présente une division biblique du monde en trois parties, en référence aux trois fils de Noé : Sem, Cham et Japhet. ▲

3.4.1 Bref historique des usages

La cartographie est ancienne, les premières tablettes d'argile connues dateraient de 2500 av. J .C (figure 3.9). Les images elles-mêmes ont précédé l'écriture de mots [Tufte 1983; Tversky 1995]. La carte consistait à l'origine à transcrire, au jour le jour, les nouvelles connaissances que l'homme avait sur le monde. Parfois, il souhaitait schématiser des croyances, d'autres fois il projetait à une échelle réduite des mesures. La carte était ainsi un outil pour percevoir un espace d'une façon globale, se repérer et se diriger à l'aide d'une métrique. Une évolution directe est la carte d'état major : on ajoute à la représentation géographique des informations sur les troupes et leurs déplacements à des fins stratégiques. Dès lors des problématiques surgissent telles que la visualisation de données dynamiques : vue du relief, points stratégiques, mouvements de troupes, etc. En parallèle des abstractions font leur apparition. La carte de T-in-O (figure 3.10) représente le monde suivant une vision biblique au moyen âge : un cercle découpé en trois secteurs.

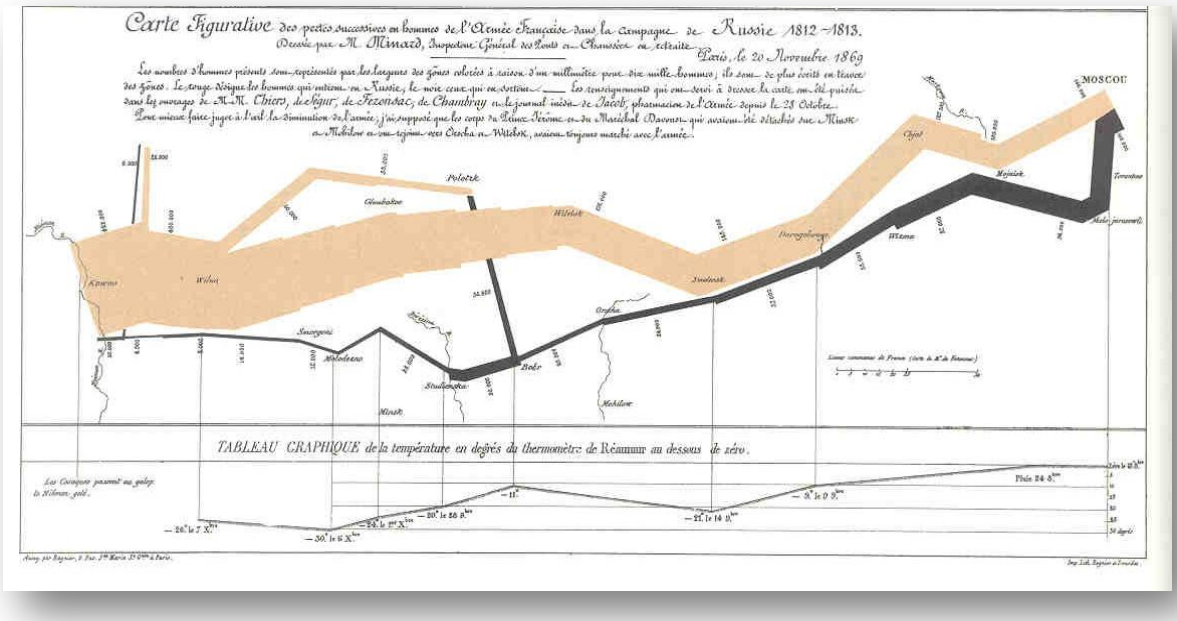


Figure 3.11 - « Carte figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812-1813 » Minard, 1861. Le trajet aller de la campagne est dessiné en rouge, le trajet retour en noir. La largeur du trait est proportionnelle aux effectifs de l'armée. En bas, la courbe indique les températures.

Les cartes thématiques font leur apparition à partir du 18° siècle. Charles Joseph Minard[1] (1791-1870) en est un précurseur et protagoniste. Sa production la plus célèbre est la carte de la campagne napoléonienne de Moscou (figure 3.11), publiée en 1869, décrite par Edward Tufte comme *« probablement le meilleur graphique statistique jamais dessiné »* [Tufte 1983]. Elle combine de multiples informations : le trajet (2 dimensions) et le sens, le temps, les pertes humaines et la température… Outre ses travaux sur des diagrammes (radiaux, à bâtons, etc.), C.J. Minard est l'auteur de cartes thématiques retraçant des flux migratoires, des importations et exportions, ou encore des cartes aidant à placer les nouveaux bureaux de postes parisiens. Un autre grand personnage fréquemment cité est John Snow[2], souvent considéré comme le père de l'épidémiologie avec la carte des cas de choléra recensés dans le quartier londonien de Soho durant l'année 1854 (figure 3.12). Elle a permis de découvrir le vecteur de la maladie, le point d'eau, et d'endiguer l'épidémie.

Figure 3.12 – Epidémie de choléra de Londres – John Snow – 1854. Chaque point correspond à une victime du choléra. La pompe à eau à l'origine de l'épidémie était au centre de la zone où les cas étaient les plus concentrés.

Dans le passé, et dès l'antiquité, les cartes ont démontré leur utilité. De nombreuses autres représentations enrichissantes sont détaillées sur le site du CSISS[3] et sur la page de Michael Friendly[4]. Plus récemment, au 20ème siècle ; la cartographie bénéficie d'un essor considérable. Cette croissance résulte de progrès techniques qui ont amélioré la conception de cartes (outils informatiques, photographies aériennes et satellites) et favorisé leur diffusion (imprimerie couleur à coût réduits, etc.). En parallèle, certains progrès ont accentué ou créé des besoins : congés payés, progrès commerciaux et marketing, analyse démographique ou épidémiologique, etc. L'informatique accentue aujourd'hui ce phénomène en facilitant l'analyse de données, rendant les cartes interactives, et créant un besoin au travers des terminaux mobiles. La cartographie s'est en effet illustrée dans l'actualité au travers du GéoPortail de l'Institut Géographique National (IGN), pilier du domaine en France, et précédemment par les services *Google Earth* et *Google Maps* proposés par le géant américain du même nom. Une institution américaine incontournable en matière de recherche et d'ingénierie cartographique est l'« *University Consortium for Geographic Information Science* » (UCGIS) [McMaster and Usery 2004]. En France, un groupe de recherche vient de se constituer, avec le groupe de travail *Cartactive* qui réuni les communautés des sciences de l'information géographique et informatique.

[1] Re-Visions of Minard, Michael Friendly, `http://www.math.yorku.ca/SCS/Gallery/re-minard.html`

[2] Ralph R. Frerichs site devoted to the life and times of Dr. John Snow
`http://www.ph.ucla.edu/epi/snow.html`

[3] Center for Spatially Integrated Social Science

[4] The Graphic Works of Charles Joseph Minard – Michael Friendly, Gallery of Data Visualization, The Best and Worst of Statistical Graphics, York University, Statistical Consulting Service and Psychology Department, `http://www.math.yorku.ca/SCS/Gallery/minbib.html`

3.4.2 Définition & motivations

3.4.2.1 Définitions générales

La spatialisation consiste à projeter une donnée dans deux ou trois dimensions, qu'elle soit à l'origine multidimensionnelle ou non. La cartographie est une spatialisation où les données sont géolocalisées (localisées par rapport à la terre). Aujourd'hui, cette notion s'étend à des données localisées ou spatialisées qui utilisent la sémantique de la cartographie (distances, chemins, frontières, etc.). La carte est l'instance de la cartographie : à partir de données cartographiques, plusieurs cartes peuvent être générées afin d'adopter et de partager des points de vue différents et nouveaux. Utilisable à plusieurs échelles, elle permet de voir émerger une connaissance plus simple à un niveau macroscopique ou de s'intéresser à une information détaillée sur une région de la carte.

Les données cartographiques peuvent être localisées dans l'anatomie d'un corps humain, dans un bâtiment (plan), dans un réseau informatique, etc. Elles peuvent aussi être localisées dans un espace multidimensionnel purement abstrait ou non localisées. C'est par exemple le cas des cartographies de l'information qui considèrent un espace ou un réseau informationnel. Dans ces derniers cas, la localisation disparaît, et seule la sémantique subsiste dans la métaphore (relief, distance, etc.).

Ces propriétés sémantiques ont été obtenues par un consensus culturel, ou sont issues du principe d'*affordance* de Gibson [Norman; Gibson 1977; Gibson 1979]: *« un objet par lui-même, influe sur son utilisation, indique comment s'interfacer avec lui »*. Cette démarche rejoint la sémiologie graphique de Jacques Bertin [Bertin 1967; Bertin 1977]. La métaphore de la cartographie qui utilise une métaphore géographique pour rendre l'espace informationnel accessible à la cognition humaine [Chalmers 1995; Skupin and Buttenfield 1997; Fabrikant and Buttenfield 2001].

Cette métaphore est qualifiée par Helen Couclelis d' *« unique et la plus riche, la source la plus systématique de sous-métaphores cohérentes pour structurer des représentations d'information complexe »*[1] [Couclelis 1998]. Employée dans de multiples contextes plus ou moins abstraits, la cartographie possède une image proche de celle que nous souhaitons produire avec notre outil. Elle favorise aussi la tendance naturelle qu'a l'utilisateur à *« aller voir ce qu'il y a derrière la prochaine colline »* [Buttenfield and Weber 1994], une stimulation de la curiosité de l'utilisateur qui correspond au problème d'analyse exploratoire de données.

La cartographie est issue des géosciences (ou sciences de l'information géographique). D'autres termes existent dans cette communauté et dont les définitions ne font pas toujours consensus. Le terme cartogramme se dissocie de la carte : la carte est le fond permettant la géolocalisation, le cartogramme est le produit qui superpose à ce fond de carte des données diagrammatiques [Andrieu 2005]. Cette notion est très proche de celle de carte thématique. Le terme géovisualisation restreint la cartographie à l'observation interactive de phénomènes spatiaux. Enfin, le terme spatialisation consiste à représenter dans un espace à 2 ou 3 dimensions des données abstraites qui ne sont pas initialement de cette nature.

3.4.2.2 Au croisement de plusieurs communautés

Les travaux théoriques en matière de cartographie sont principalement issus des géosciences. La carte est cependant visuelle, et a pour objectif de faire sens sans nécessiter d'apprentissage important : l'utilisateur sait ce qu'est une carte et comprend sa sémantique de façon relativement intuitive, éventuellement en se référant à une légende. Elle poursuit alors des objectifs similaires à ceux de la visualisation d'information, synthétisés par Edward Tufte comme suit :

[1] *« … single richest, most systematic source of coherent submetaphors for structuring complex information representations. »*

Clarity and excellence in thinking is very much like clarity and excellence in the display of data.[1] [Tufte 1983]

Il rejoint le principe de congruence et l'ancien adage attribué à Confucius *« Une image vaut mieux que mille mots »*. Ensuite, il définit *l'excellence graphique* :

Graphical excellence is the well-designed presentation of interesting data – a matter of substance, of statistics, and of design. Graphical excellence consists of complex ideas with clarity, precision, and efficiency. Graphical excellence gives the viewer the largest number of ideas in the shortest time with the least ink. [2]*[Tufte 1983]*

Il propose dans ses ouvrages des outils et des méthodes objectives permettant de construire et d'évaluer cette excellence [Tufte 1983; Tufte 1990; Tufte 1997]. La plus connue est certainement le ratio encre/donnée :

$$ratio = \frac{encre\ utilisée\ pour\ les\ données\ non\ redondantes}{encre\ totale\ utilisée}$$

Il prodigue de nombreux autres conseils pratiques, et introduit d'autres notions comme l'intégrité des données. Une seconde référence issue du domaine des géosciences est l'œuvre de Jacques Bertin [Bertin 1967; Bertin 1977]. Il se concentre sur la « sémiologie graphique » (ou sémiotique) : il étudie comment *« la graphique »* fait sens sans définition et apprentissage préalable. A partir des connaissances physiologiques de la perception visuelle et de la cognition, il explique comment utiliser efficacement les signes.

Comme le montrent ses travaux qui se basent sur des critères de quantité d'encre, etc. E. Tufte s'est intéressé dans un premier temps à des supports papier. L'informatique a progressé dans le quotidien du cartographe et des utilisateurs de cartes. La conception de carte se fait à l'aide de logiciels et de systèmes d'information géographique ; les cartes deviennent interactives, cliquables, zoomables, et animées. Le cartographe est alors amené à utiliser les outils informatiques de visualisation d'information. En parallèle, la communauté informatique de la visualisation s'intéresse aux mêmes aspects. Jock Mackinlay et Colin Ware prennent en compte par exemple les caractéristiques physiologiques et cognitives de la perception visuelle pour établir des recommandations pratiques dans la conception de visualisations [Mackinlay 1986; Ware 2000]. Ces deux disciplines sont maintenant réunies : elles partagent des outils, des techniques et pratiques, des utilisateurs et des fondements théoriques. Ces rapprochements pluridisciplinaires sont particulièrement bien illustrés par l'UCGIS ou encore le groupe de travail du CNRS en France. Les travaux de Sara Fabrikant et de ses collaborateurs en sont représentatifs [Fabrikant and Buttenfield 2001].

La carte est omniprésente : elle est disponible sur les sites touristiques, dans les Pages Jaunes ou par exemple sur les sites d'information concernant les élections passées. Elle envahit les terminaux mobiles (assistants de poche, téléphones, GPS pour les routes, les voies navigables et les randonnées, etc.). De nouvelles questions se posent avec la diversité des usages, des utilisateurs et des supports (taille de l'écran, figée ou animée, etc.). Que devient la légende ? Comment l'utilisateur est-il capable de personnaliser sa carte ? Ce sont ces questions auxquelles tentent de répondre communément informaticiens et géographes.

Enfin, soulignons l'interpénétration de ces deux communautés. Un exemple anecdotique est la mesure d'Horton et Strahler employée en géo-hydrologie [Strahler 1952]. Utilisée initialement

[1] *La clarté et l'excellence dans la pensée sont fortement similaires à la clarté et l'excellence dans l'affichage des données.*

[2] *« L'excellence graphique est la présentation bien conçue de données intéressantes – un problème de substance, de statistique et de conception. L'excellence graphique consiste en la communication d'idées complexes avec clarté, précision et efficacité. L'excellence graphique donne au lecteur le plus grand nombre d'idées en un temps minimum avec un minimum d'encre. »*

pour calculer la largeur de la représentation d'un cours d'eau sur une carte (plus généralement sa taille), cette mesure a été reprise pour la visualisation d'arbres binaires et n-aires [Auber, Delest et al. 2004.].

3.4.2.3 Le rôle du support graphique

Nous avons déjà cité les adages *« Une image vaut mieux que mille mots »* de Confucius repris par Napoléon *« Un dessin vaut mieux qu'un long discours »*. Au-delà de simples adages, il existe de multiples motivations pour visualiser des données.

La vision est la fonction la plus développée dans le cortex cérébral de l'homme ; elle en monopoliserait plus de la moitié. La faculté de vision est le résultat d'une évolution qui s'est faite en partie au détriment de notre système olfactif. Elle est notre principale source de perception. Plus généralement, la perception permet de traiter en parallèle une grande quantité d'informations. Au contraire, notre réflexion est séquentielle, et nécessite l'utilisation de la mémoire à court terme dont la capacité est restreinte[1]. Enfin, la présentation visuelle de l'information met en œuvre une activité distincte dans la mémoire. La vision permet donc de réduire la charge cognitive d'une tâche et d'améliorer l'activité mnémonique dans cette tâche (rétention et rappel). Des mécanismes de préattention permettent par ailleurs de détecter des mouvements et des changements. Il est important de prendre en compte les mécanismes de perception et de cognition dans la communication et l'analyse de l'information. C'est ce que résume la phrase de Ben Schneiderman :

> *«The eye is the best way for the brain to understand the world around us. »*[2]
>
> *[Shneiderman 1987]*

L'objectif d'une visualisation (dans une carte ou non) est donc de faciliter l'analyse, la compréhension et le partage de données complexes [Tufte 1990]. Arnheim et MacEachrean et Card utilisent le terme de *« raisonner visuellement »* [Arnheim 1969; MacEachrean 1995; Card, Mackinlay et al. 1999]. McCormick propose la définition suivante :

> *Visualization is a method of computing. It transforms the symbolic into the geometric, enabling researchers to observe their simulations and computations. Visualization offers a method for seeing the unseen. It enriches the process of scientific discovery and fosters profound and unexpected insights. In many fields it is already revolutionizing the way scientists do science.* [3]

D'autres définitions existent dont nous recommandons la synthèse proposée par le WikiViz. Par ailleurs trois principaux objectifs de la visualisation d'information y sont décrits : la communication (présentation, mémorisation, argumentation, etc.) [Röber 2000], l'analyse exploratoire de données (qui englobe selon eux l'aide à la décision) et l'analyse confirmatoire (vérification d'hypothèse). Ces deux dernières démarches correspondent à celles des biologistes avec lesquels nous collaborons : dans un premier temps, ils analysent au niveau macroscopique du génome entier l'expression des gènes (puces à ADN). Une fois qu'ils ont isolé quelques gènes d'intérêt, ils vérifient leurs hypothèses par exemple à l'aide de QRT-PCR.

Enfin, ils définissent la visualisation d'information par son rôle essentiel de mise en relation d'éléments de données (distance, liens, taille, etc.).

---

[1] On considère généralement qu'elle contient 7±2 *« cases »*

[2] *« Les yeux sont le meilleur moyen pour le cerveau de comprendre le monde qui nous entoure »*

[3] *La visualisation est une méthode informatique. Elle transforme le symbolique en géométrique, permet aux chercheurs d'observer leurs simulations et calculs. La visualisation offre une méthode pour voir le « non vu ». Elle enrichit le processus de découverte scientifique et favorise des percées profondes et inattendues. Dans de nombreux domaines, elle révolutionne déjà la façon dont les scientifiques font des sciences.*

3.4.3 Approche théorique

3.4.3.1 Fondements de la cartographie (et de la spatialisation)

> *« Michel et Jean sont au téléphone, et Michel donne à Jean les directions à suivre pour venir chez lui. Jean n'a jamais été chez Michel auparavant et ne connaît pas très bien sa géographie. Pour mieux comprendre la description de Michel, Jean commence à dessiner un croquis basé sur les explications orales de Michel. » [Klippel, Lee et al. 2005]*

Dans cette exemple, Jean crée une représentation spatiale conceptuelle basée sur ses connaissances partielles et prenant en compte les différentes contraintes élémentaires (croisement, pont, etc.) [Klippel and Tappe 2001]. La carte est créée à partir d'une représentation mentale, en accord avec une connaissance partagée. Mark Johnson détermine à partir d'expériences sensorimotrices sept schémas mentaux dans l'interprétation de spatialisation : le contenant, la surface, la proximité, la verticalité, le lien, le chemin et la centralité [Johnson 1987]. La mise en œuvre de ces schémas dans une spatialisation peut être sémantique ou structurelle (topologique) [Fabrikant and Buttenfield 2001]. Ces approches ne sont pas contradictoires et peuvent n'être que partiellement mises en œuvre.

Dans [Fabrikant and Buttenfield 2001], les auteurs introduisent trois sortes d'espaces d'information et d'approches de la spatialisation : l'espace géographique, cognitif et Benediktin (en référence aux travaux de Michael Benedikt sur le cyberespace [Benedikt 1991]). L'espace géographique est ancré dans les lois physiques du monde réel. Il contient ainsi un *continuum* d'échelles allant du microscopique au macroscopique. On y retrouve un système de coordonnées (longitudes et latitudes ou les points cardinaux) muni d'une loi interdisant la présence de deux entités distinctes en un même endroit de l'espace. La gravité introduit la notion d'altitude, de sol, éventuellement de sous-sol. Ces notions introduisent des notions de points de vues différentes : sous-marin, aérien, portée, etc.

L'espace cognitif fait référence à la conceptualisation de l'espace et à la représentation mentale qu'en a l'utilisateur. Dans ce cadre, la notion d' « absolu » disparaît et plus généralement les distances, coordonnées, et orientations ne sont que faiblement privilégiées [Mark and Frank 1991]. Les valeurs sont alors relatives, moins précises, et subjectives. Ces représentations mentales sont construites par les expériences de l'utilisateur. La connaissance de l'espace est l'une des premières connaissances acquises par l'homme [Taylor and Tversky 1996]. Elle se construit non pas par la perception de l'espace, mais par les interactions avec cet espace [Golledge and Stimson 1987]. D'après [Tversky 1995] et [Golledge and Stimson 1987], l'organisation de cet espace est hiérarchique, par prépondérance, par fonction ou par association entre des éléments du haut et du bas de la hiérarchie. Dans le cas de connaissances bioinformatiques, c'est dans cet espace cognitif que se positionne notre problématique.

Klippel décrit la cartographie en isolant les méthodes de construction de cartes suivant deux approches distinctes : une approche basée sur les données (*« Data Driven approach »*) et une approche conceptuelle (*« Cognitive Conceptual Approach »*). L'approche cognitive débute par une représentation mentale qu'on tente d'affiner schématiquement, elle se base sur l'espace cognitif de [Fabrikant and Buttenfield 2001]. L'approche basée sur les données, au contraire, part de ces données géographiques et physiques absolues et en abstrait incrémentalement la représentation. Cette seconde approche se base donc sur l'espace géographique [Fabrikant and Buttenfield 2001]. Dans notre cas, les données sont déjà abstraites à un niveau conceptuel de par les annotations, la symbolique des réactions chimiques, etc.

Quelle que soit l'approche envisagée, tout le monde s'accorde à considérer qu'une carte, comme tout autre objet manipulé, doit être adaptée à l'utilisateur et l'utilisation [Barkowsky and Freksa 1997; Freksa 1999; Muehrcke, Muehrcke et al. 2001; Klippel, Lee et al. 2005]. Dans une perspective informatique et dynamique, on rejoint le mantra de Ben Schneiderman *« Overview*

first, zoom and filter, then details-on-demand »[1] *[Shneiderman 1996].* A cela on peut ajouter la nécessité d'une adaptation de la vue et des données au contexte du domaine et de l'utilisation.

3.4.3.2 Propriétés des schémas spatiaux

M. Johnson a mis en évidence 7 propriétés graphiques adaptées par S. Fabrikant à la cartographie [Johnson 1987; Fabrikant and Buttenfield 2001] : le contenant, la surface, la proximité et l'éloignement, le verticalité, le chemin, le lien et le centrage. Nous proposons de reprendre ces propriétés afin de montrer comment elles peuvent être mises en œuvre de façon variée dans des cartes, alternativement ou conjointement. Il est important de noter que ces variations sont liées : en modifiant les surfaces, on modifie certaines coordonnées, on rapproche deux éléments, et on les associe plus facilement.

Conteneur et surface

Les conteneurs ont un intérieur et un extérieur, avec une frontière délimitant les deux. Cette notion de frontière dissocie la surface, continue, du conteneur. Comme usage géographique, on peut, par exemple, opposer la carte géopolitique qui sépare les pays en dessinant leurs frontières et les autres cartes qui font état de températures, relief, pollution, météorologie par exemple sans afficher les frontières. La figure 3.13 en est un exemple issu du domaine médical. Dans le domaine médical, l'imagerie médicale donne une vision non cloisonnée du cortex alors que le manuel médical le décompose en lobes.

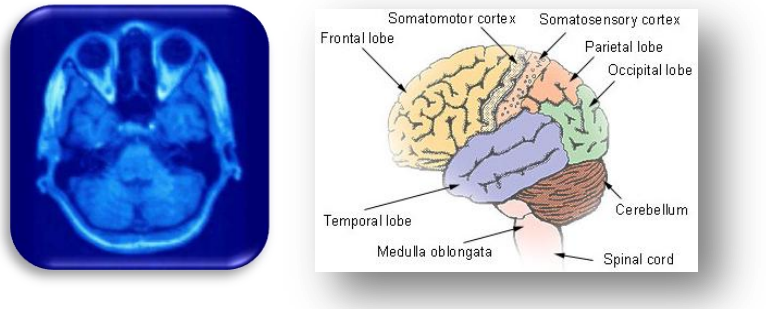
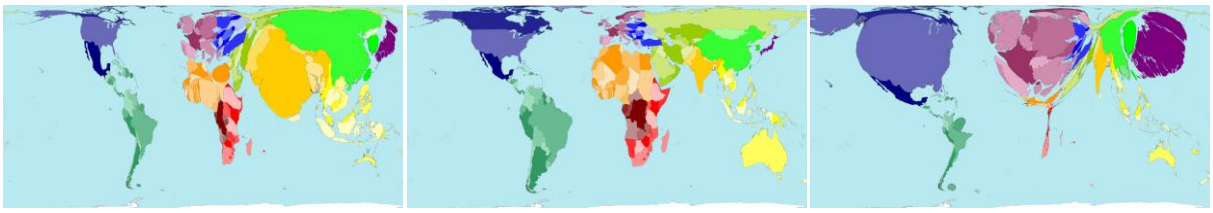


Figure 3.13 –.La visualisation du cerveau peut être découpée en régions (à droite) ou être perçue comme une surface continue.

La surface est une variable. Certaines cartes géopolitiques choisissent une échelle (ou un gradient) de couleurs pour représenter une variable quantitative. D'autres choisissent au contraire de modifier la surface afin de montrer un déséquilibre. C'est une démarche utilisée actuellement dans le contexte de la consommation, des revenus ou de la pollution afin d'obtenir un effet plus convaincant (figure 3.14).



Population *Superficie* *Revenu national*

Figure 3.14 – La carte géopolitique du monde peut être déformée par la surface afin de faire apparaître un déséquilibre. Au centre, la carte dessine la surface d'un pays en fonction de sa superficie. A gauche on peut voir les proportions de populations dans le monde, à droite celle du revenu national. Chaque carte affiche donc une information, la comparaison fait émerger une information supplémentaire.

[1] D'abord un aperçu global, zoom et filtre, puis des détails à la demande.

Une distance *d* sur un ensemble *E* est une application $d : E \times E \rightarrow R^+$ telle que :

$\forall x, y \in E \quad d(x,y) = d(y,x)$ *(symétrie)*

$\forall x, y \in E \quad d(x,y) = 0 \Leftrightarrow x = y$ *(séparation)*

$\forall x, y, z \in E \quad d(x,z) \leq d(x,y) + d(y,z)$ *(inégalité triangulaire)*

Figure 3.15 – Définitions mathématiques de la distance

Proximité & Eloignement : la distance

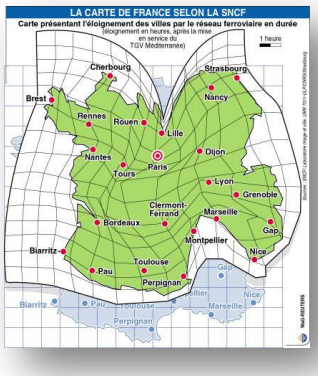
Fabrikant et Buttenfield mentionnent la proximité ou l'éloignement, et non la distance. En effet, l'œil n'est pas une caméra. Il ne permet pas une mesure objective des distances, surfaces ou couleurs par exemples, il permet de comparer des éléments :

x est plus gros que y, *x est plus foncé que y,* *etc.*

Concernant la distance, le même principe est appliqué : la distance permet d'obtenir plusieurs informations :

x est plus proche de y que z, *x est plus haut que y,* *les plus proches voisins de x sont y et z,*
x et y sont éloignés, *x, y et z sont regroupés dans la même région*

La distance est une mesure mathématique qui a notamment la propriété d'inégalité triangulaire que ne possède pas la mesure de similarité (figure 3.15). La distance euclidienne est le plus souvent employée, mais on peut en utiliser d'autres (distance de Manhattan, de Minkovsky, etc.). La phylogénétique, par exemple, propose plusieurs mesures de distance, de similarité et de score concernant l'alignement entre séquences [Guindon 2003] qui peuvent être visualisées dans des dendrogrammes. Un exemple courant est celui de la carte du territoire français exprimé en fonction du temps qui sépare les gares ferroviaires de Paris (figure 3.16). Ce changement de métrique n'est pas anodin puisqu'il ne respecte plus l'inégalité triangulaire.



	Distance à vol d'oiseau	Durée d'un trajet en train
Paris-Lyon	390 km	1h54
Paris-Le Mans	184 km	0h56
Lyon-Le Mans	**430 km**	6h23
Lyon-Paris + Paris Le Mans	574 km	**2h50**

Figure 3.16 – Carte de France proposée par la SNCF basée sur une dissimilarité en temps via le réseau ferré de France. Ne respectant plus l'inégalité triangulaire, il ne s'agit plus d'une mesure de distance d'après la définition mathématique de cette dernière.

Verticalité : dimensions

La problématique abordée par cette propriété est celle des dimensions. Dans les cartes géographiques précédentes, on disposait de deux dimensions : la longitude et la latitude. Dans

de nombreux cas, on utilise une troisième dimension. Physiquement, elle représente la hauteur d'un bâtiment, le relief d'une montagne, la profondeur sous-marine, etc. Dans le contexte abstrait d'un espace informationnel, elle représentera par exemples des sujets importants, actifs, des régions denses, etc.

Figure 3.17 – Ces cartes représentent des espaces informationnels obtenus à partir de cartes de Kohonen. En haut, ThemeScape Propose une visualisation en 2D avec des lignes de niveaux. A droite, S. Fabrikant propose un véritable univers virtuel en 3D.

Il existe plusieurs approches pour représenter cette troisième dimension. De la même façon que l'IGN le fait dans les cartes routières, ThemeScape utilise une vue de dessus conjointement à des lignes de niveaux et dégradés de couleurs (figure 3.17). Il est aussi possible d'utiliser une visualisation dans 3 dimensions (figure 3.17) [Skupin and Buttenfield 1997]. Cette troisième dimension induit des phénomènes d'occlusion et une vue partielle de l'espace. Elle n'est pas simple à utiliser, notamment pour les générations qui ne sont pas « nées » avec les jeux vidéos en 3D et sont peu utilisées. Leur usage se limite à des problèmes où la nature des données et la tâche à réaliser s'y prêtent bien. Enfin, il existe une dernière approche par fois nommée « 2,5D ». Au sein d'une visualisation deux dimensions, les objets sont d'autant plus gros qu'ils sont bas. Cela produit une impression de recul et de perspective, mais ne s'agit pas de 3D : il y a pas d'intersection de volumes, de rotation ou retournement, etc. Les *Data Mountains* développées par Microsoft ont introduit cette notion en l'appliquant à la gestion de signets [Robertson, Czerwinski et al. 1998] (figure 3.18).

Figure 3.18 – Data Mountains (Microsoft) : plus une capture est en haut de l'écran, plus elle est petite, et plus elle est rarement utilisée. Cela procure une impression de profondeur. Il n'y pas d'intersection entre les éléments, de possibilité de changer d'angle de vue, d'appliquer des rotations à des objets et ou de les retourner. On parle donc de 2,5D et non de 3D.

Chemins et liens

Les liens sont des éléments importants dans la visualisation et la cartographie. Une distinction est faite entre le chemin et le lien dans [Fabrikant and Buttenfield 2001] : le lien est une relation entre deux éléments représentée par une connexion, une adjacence, etc. Le chemin est perçu comme un flux avec une origine et une destination.

Nous adoptons sur ce point une position plus proche de la vision mathématique du graphe : le chemin est une succession de liens. En effet, le chemin n'est pas toujours orienté (carte routière, plan de métro, schéma d'un réseau social, etc.) et le lien l'est parfois (relations sémantiques, etc.). Le chemin ou le lien n'ont pas toujours d'intérêt, et l'information recherchée au travers de leur visualisation est parfois liée à la structure du graphe : hiérarchie, treillis, planarité, régions denses, diamètre, connexité, présence de cycles, etc.

Les exemples de cartes de ce type sont nombreux. Dans le contexte de la biologie, les plus courants sont les voies métaboliques et voies de signalisation, les réseaux de régulation de gènes, et les réseaux d'interactions.

Le centrage

Un dernier élément abordé est la notion de centre et de périphérie. L'utilisation d'un centre implique généralement que les dimensions perdent leur sémantique. La distance au centre permet d'ordonner les éléments d'information en fonction d'un degré d'intérêt.

Des grandes variations

Nous venons de montrer les différentes propriétés porteuses de sémantique dans la carte, mais source de variations. Il existe bien d'autres facteurs : symboliques, sémiotiques, culturels sur lesquels peut jouer le cartographe. Dans ce dernier paragraphe, nous illustrons simplement le propos en montrant comment une propriété peut être exacerbée au point de poser la question de la frontière entre cartographie et visualisation.

Les trois cartes de la figure 3.19 donnent une prépondérance à la surface. La première *(a)* simplifie le contour de chaque pays. Chaque contour est formé d'angles droits uniquement, la plupart ne sont que des rectangles. La surface est liée à la taille de la population. La disposition respecte cependant globalement la géographie de notre planète. La seconde carte *(b)* montre comment on peut réduire les continents et leurs pays à des simples « carrés » dont la taille est proportionnelle à la population. Cette visualisation est appelée TreeMap [Shneiderman 1992]: l'objectif n'est plus alors de percevoir notre monde d'un point de vue géographique mais

uniquement analytique par rapport à des données statistiques. La carte du marché boursier *(c)* de SmartMoney permet par exemple de localiser des secteurs à différents niveaux qui progressent, et éventuellement des évènements accidentels dans ces secteurs. Dans cette capture, on voit par exemple que le secteur des télécommunications est globalement en progrès sur l'année, sauf concernant AMD qui accuse une baisse de 29%.

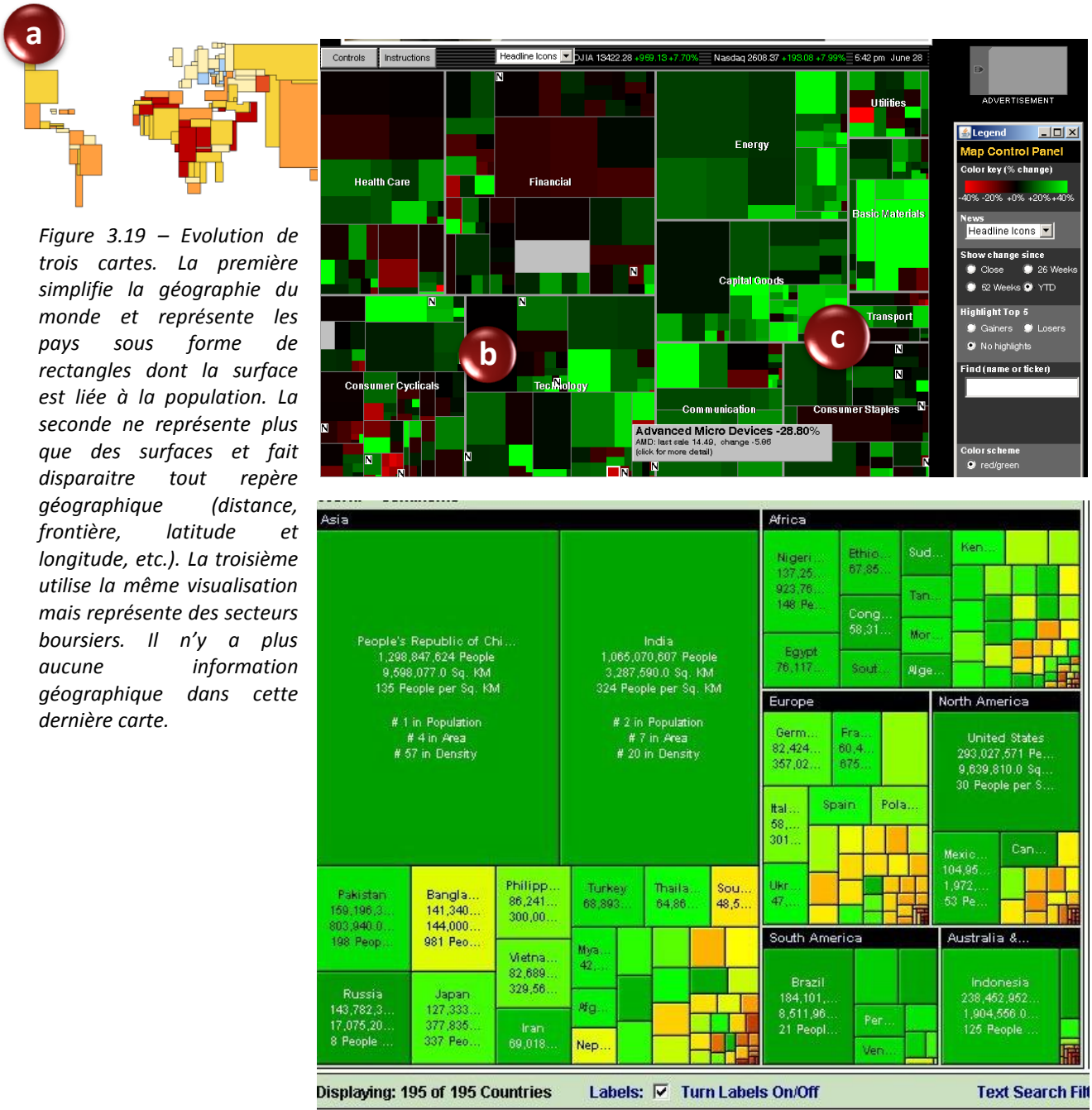


Figure 3.19 – Evolution de trois cartes. La première simplifie la géographie du monde et représente les pays sous forme de rectangles dont la surface est liée à la population. La seconde ne représente plus que des surfaces et fait disparaitre tout repère géographique (distance, frontière, latitude et longitude, etc.). La troisième utilise la même visualisation mais représente des secteurs boursiers. Il n'y a plus aucune information géographique dans cette dernière carte.

La question que nous posons est où s'arrête la cartographie ? S'agit-il réellement de trois cartes ? Les distorsions et les évolutions séparent petit à petit le planisphère que nous connaissons de la visualisation de données boursières. Dans la première carte (a), la géographie est grossièrement respectée. Il localise par exemple simplement les différents continents, la France et l'Angleterre. Mais par contre, la chine apparait au dessus de l'Inde, à l'endroit même ou la Russie est habituellement disposée. La confusion est probable. Faire perdre à l'utilisateur ses repères est cependant la volonté de cette carte, afin de l'obliger à regarder l'information sous un jour nouveau. Dans cette carte, une grande partie de la sémantique reste présente : dimensions, frontières communes à certains pays, représentation des océans, etc. Dans la seconde figure (b), il s'agit toujours de données géographiques, mais la visualisation fait totalement disparaitre tout

repère. Seule reste présente l'appartenance d'un pays à un continent. Cette visualisation oblige l'utilisateur à voir le monde en dehors de toute géographie et à traiter les données avec plus d'objectivité. Il ne se concentre alors plus que sur une seule variable : la surface. Les TreeMaps sont adaptées à la visualisation d'une hiérarchie générée par une relation d'inclusion et de « partie-tout ». Contrairement à la première figure, lorsque l'utilisateur regarde cette visualisation à un niveau macroscopique, rien ne lui laisse deviner qu'il s'agit de données géographiques. Pour distinguer cette carte de SmartMoney, il doit commencer par lire le contenu des cases et découvrir la sémantique des couleurs, surfaces et conteneurs. Il n'y a plus de chemin de distance, de frontières et d'adjacence, d'océan, de latitude ou de longitude.

La représentation habituelle du métro a été reprise pour dessiner les courants musicaux (figure 3.20). L'aspect artistique et le rendu esthétique sont intéressants. Cependant, nous nous interrogeons sur le bienfondé d'une telle représentation : quelle est la signification de la tamise ? Que représente une station car on ne peut définir un artiste de façon ponctuelle ? Que représentent les boucles, les intersections avec ou sans stations, et les virages ? Que représentent les dimensions verticales et horizontales ? Que représentent les distances entre les stations ? Dans cette cartographie, l'auteur présente une information particulièrement complexe alors que cette complexité ne sert pas l'utilisation faite de la carte.

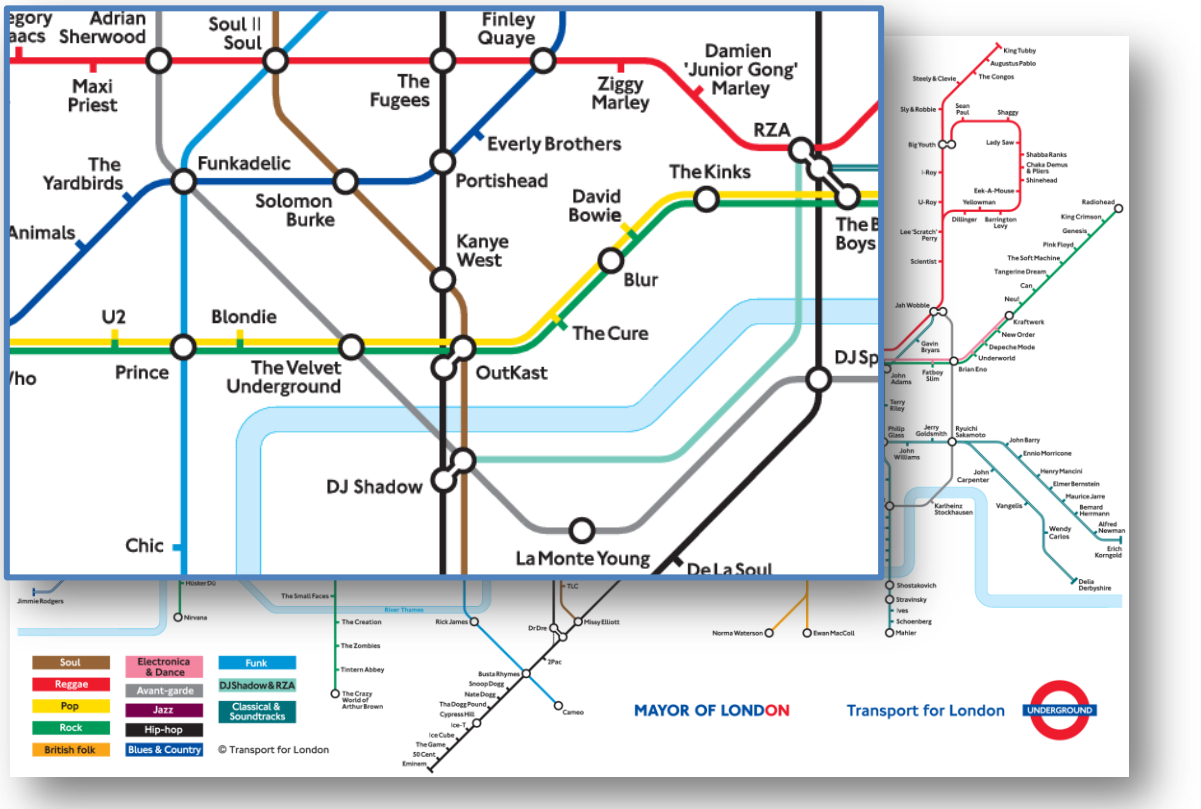


Figure 3.20 – Cartographie des courants musicaux s'inspirant de la carte du métro londonien.

3.4.4 Cartographie par l'usage

Après avoir défini les principaux éléments théoriques de la carte et en avoir montré quelques illustrations, nous proposons maintenant de nous intéresser aux usages informatiques et bioinformatiques de la carte. Dans un premier temps, ces usages sont présentés suivant les différentes natures de données qui sont cartographiées. Dans un second temps, nous détaillons quelques projets principaux relatifs au domaine biomédical.

3.4.4.1 Topologie et nature des données

Les données peuvent être initialement sous forme de graphe ou de données multivariées. On peut spatialiser des données multidimensionnelles en se basant sur les coordonnées (analyse factorielles, carte de Kohonen, etc.) ou à partir de la matrice de distances (modèle de force, etc.). Nous avons par ailleurs mentionné que l'intégration d'un graphe dans un espace multidimensionnel met en jeu des techniques mathématiques complexes et difficiles à évaluer. Au contraire, la génération d'un graphe à partir de données multidimensionnelles se fait assez facilement en générant un graphe complet ou les arêtes ont une longueur correspondant à la matrice de distance de l'espace vectoriel.

Dans la suite nous présentons des exemples de cartographie en considérant la nature des données suivant un point de vue fonctionnel : des mots et concepts, des documents, des acteurs, etc.

Cartographie lexicale

Différents outils permettent de visualiser des graphes de mots. Les applications les plus courantes sont la navigation dans une ontologie, le parcours d'une taxonomie pour accéder à ses signets, la recherche documentaire, ou la supervision de bases lexicales par des linguistes.

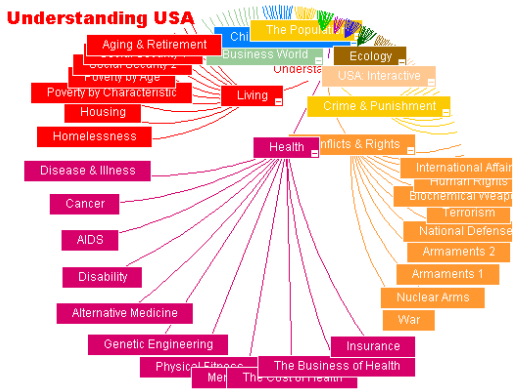


Figure 3.21 – StarTree Viewer d'InXight propose de visualiser des arbres à partir d'une géométrie hyperbolique.

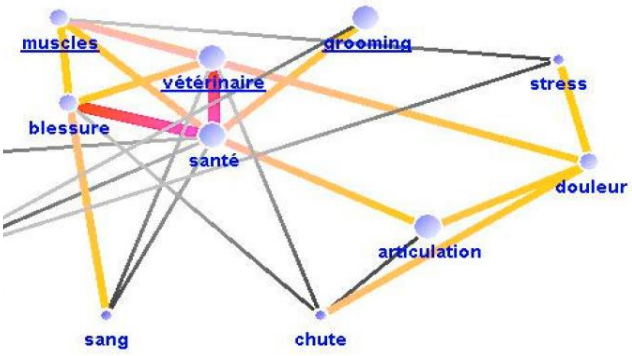


Figure 3.22 – VicoText – Visualisation du contexte d'un terme pour la recherche d'information

En matière de supervision de réseaux lexicaux, nous avons notamment connaissance de deux projets : Hyperlex [Véronis 2004] et l'atlas de la synonymie développé par Sabine Ploux [Ji and Ploux 2003]. Jean Véronis utilise H3Viewer basée sur la géométrie hyperbolique [Munzner 2000]. Sabine Ploux utilise une analyse factorielle pour la spatialisation [Escofier and Pagès 1988], associée à une méthode de regroupement. On retrouve de multiples applications de la géométrie hyperbolique commercialisée par InXight (filiale de Xerox) dans le logiciel Star Tree Viewer (figure 3.21). Mathieu Lafourcade [Lafourcade, Prince et al. 2002] suit une approche vectorielle descendante de Salton [Salton and Buckley 1987] où chaque dimension correspond à un concept du Thésaurus Larousse [Pechoin 1999]. Abdenour Mokrane [Mokrane 2006] propose d'utiliser un réseau de mots clés pour raffiner la recherche documentaire (figure 3.22). Reena Shetty construit des réseaux sémantiques automatiquement à partir d'un noyau généré par l'expert (les réseaux sémantiques étendus) qu'elle utilise dans différents contextes dont la recherche d'information [Shetty, Riccio et al. 2006].

Ces cartographies ne sont par ailleurs pas toujours des mots, mais des termes ou concepts. On peut alors mentionner les outils de visualisation et de navigation dans des terminologies et ontologies. Parmi eux, les approches graphiques reposent notamment sur UML (avec l'éditeur DUET) et sur les graphes conceptuels (avec l'éditeur Cogitant)[Sowa 1984; Genest and Salvat 1998]. La carte heuristique (traduction de « *mind map* ») [Le Bihan, Deladrière et al. 2004] propose de structurer des idées en arbres. La production et l'usage sont généralement manuels et il s'agit plus d'une méthodologie de travail en groupe.

Réseaux sociaux

Les sciences sociales s'intéressent depuis longtemps aux réseaux sociaux. L'informatique est sollicitée pour des problématiques de formalisation, de génération, d'analyse ou de visualisation de ces réseaux notamment. Du point de vue des sciences humaines et sociales, les réseaux sociaux servent à étudier les comportements sociaux, alors que du point de vue de l'informatique, au même titre que les graphes *petit monde*, ils sont devenus une topologie de graphe avec certaines propriétés. Nous ne connaissons pas d'approche pluridisciplinaire dans ce domaine. Dans une entreprise, on s'intéresse aussi à la cartographie des connaissances et des compétences. Il s'agit de schématiser les différentes ressources afin d'orienter des décisions stratégiques (politique de recrutement, etc.).

Les réseaux sociaux ont une topologie caractéristique : le graphe possède un degré moyen faible, mais certains nœuds appelés *« hubs »*, sont connectés à un grand nombre d'autres. Ces graphes sont réputés pour leur difficulté à être dessinés. Certaines tentatives de regroupement automatique ont donc été mises en œuvre. Par exemple : Vizter est une application basée sur la boîte à outils Prefuse [Heer and Boyd 2005] (figure 3.23), New Media utilise une carte de Kohonen pour cartographier les relations stratégiques et financières de 500 grandes entreprises du domaine du Media. Auber et al. s'intéressent aux participations des acteurs dans des films recensées par d'IMDB [Auber, Chiricota et al. 2003]. Enfin, Imane Anoir utilise l'environnement MolAge pour étudier la « bulle de confiance » au sein du programme ToxNuc-E [Anoir, Penalva et al. 2005; Crampes, Ranwez et al. 2006].

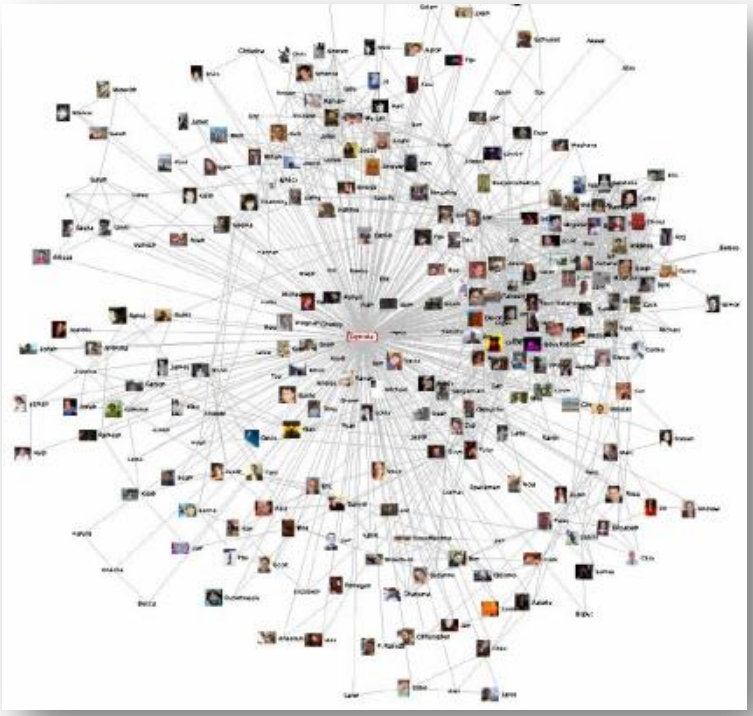


Figure 3.23 – Vizster, visualisation de réseaux sociaux basée sur Prefuse. [Heer and Boyd 2005; Heer, Card et al. 2005]

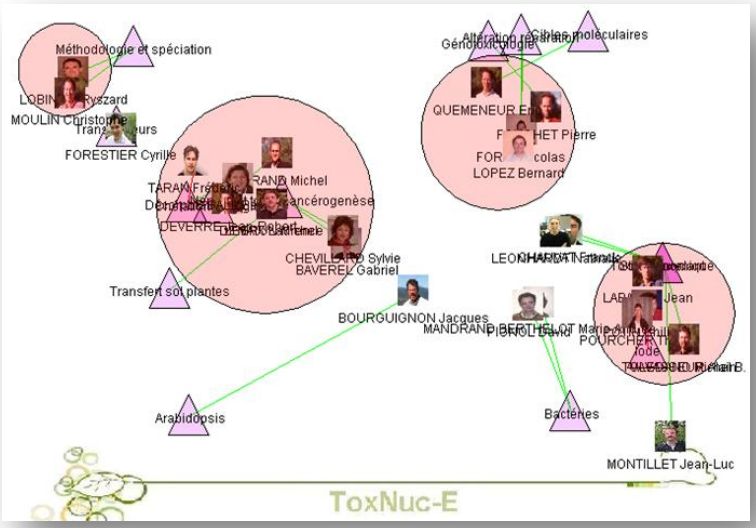
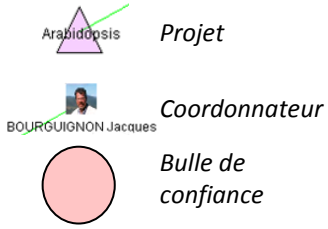


Figure 3.24 – Vizster, visualisation de réseaux sociaux basée sur Prefuse. [Heer and Boyd 2005; Heer, Card et al. 2005]



Cartographie documentaire, espaces d'information

Les espaces d'information représentent un grand nombre de contributions. Les choix de représentation de l'information permettent de catégoriser les visualisations employées. Certaines représentations vectorielles donnent lieu à des projections en deux ou trois dimensions, d'autres travaux concernent l'utilisation de cartes auto-organisatrices de Kohonen [Kohonen 1990]. Kartoo est un moteur de recherche qui représente des termes, titres, locutions ou noms de sites. Il met en évidence des relations entre les pages, des poids entre les sites, propose des raffinements, etc. TouchGraph [Shapiro 2003] propose une visualisation à partir d'un modèle de force [Eades 1984] et basé sur les résultats de Google. Une version appelée HubMed réalise la même chose en interrogeant PubMed.

L'approche des Topic Maps ressort en particulier dans ce domaine. Elle considère différents niveau d'un graphe, notamment les sujets (concepts) et leurs occurrences ou instances dans les documents. Les Topic Maps sont devenus une norme ISO [Michel Biezunski and Newcomb 1999; XTM 2001; Park and Hunting 2002]. B. Legrand en fait un état de l'art et propose une visualisation en *« camemberts »* imbriqués [Le Grand 2001]. Les Topic Maps sont parfois appelés *« GPS de l'univers d'information ».*

Une approche « localisée » a été mise en œuvre dans le projet Zoomit [Pook and Lecolinet 2002] (figure 3.25). L'utilisateur navigue dans les rayonnages de la bibliothèque de l'Ecole des Mines de Nantes en utilisant un zoom optique et logique : plus on grossit la vue, plus l'information est détaillée. Enfin, quelques approches consistent à cartographier le contenu d'un site Web.

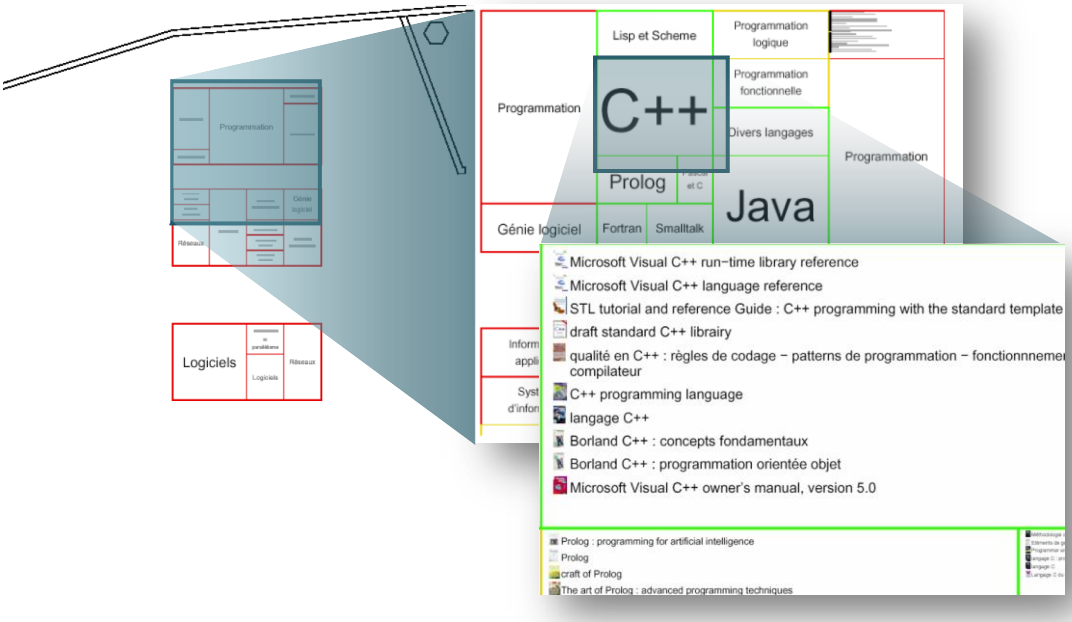


Figure 3.25 – Cartographie de la bibliothèque de l'Ecole des Mines de Nantes. Ces trois captures retracent le zoom progressif à la fois optique et logique mis en œuvre.

3.4.4.2 Exemples d'applications aux données biomédicales

Cartographie du génome

Ce projet est, de loin, le plus renommé [Danchin 1998]. La visualisation n'est pas importante dans ce projet. L'effort consiste essentiellement à séquencer des génomes entiers et à annoter les gènes identifiés, localisés sur les chromosomes. Bien que l'interaction ne soit pas l'essence du projet, des navigateurs permettent de parcourir ces génomes. Nous en avons présentés dans la section (3.2.2 page 81).

Réseaux biologiques

Par réseau biologique nous entendons des réseaux reliant des entités biologiques : gènes, protéines, composés chimiques, etc. Les travaux en la matière sont très fournis. Des outils comme PubGene [Jenssen, Laereid et al. 2001] (figure 3.27) et BioBiblioMetrics [Stapley and Benoit 2000] par exemple s'inscrivent dans la lignée des travaux de D.R. Swanson [Swanson 1986; Pierret and Boutin 2004]: un graphe est construit par l'extraction de relations statistiques (cooccurrences, etc.) à partir des résumés d'articles de PubMed. L'utilisateur fourni une liste de gènes (par exemple ceux présents sur une puce à ADN), et le système retourne un sous graphe qui contient toutes les relations entre les gènes provenant de cette littérature.

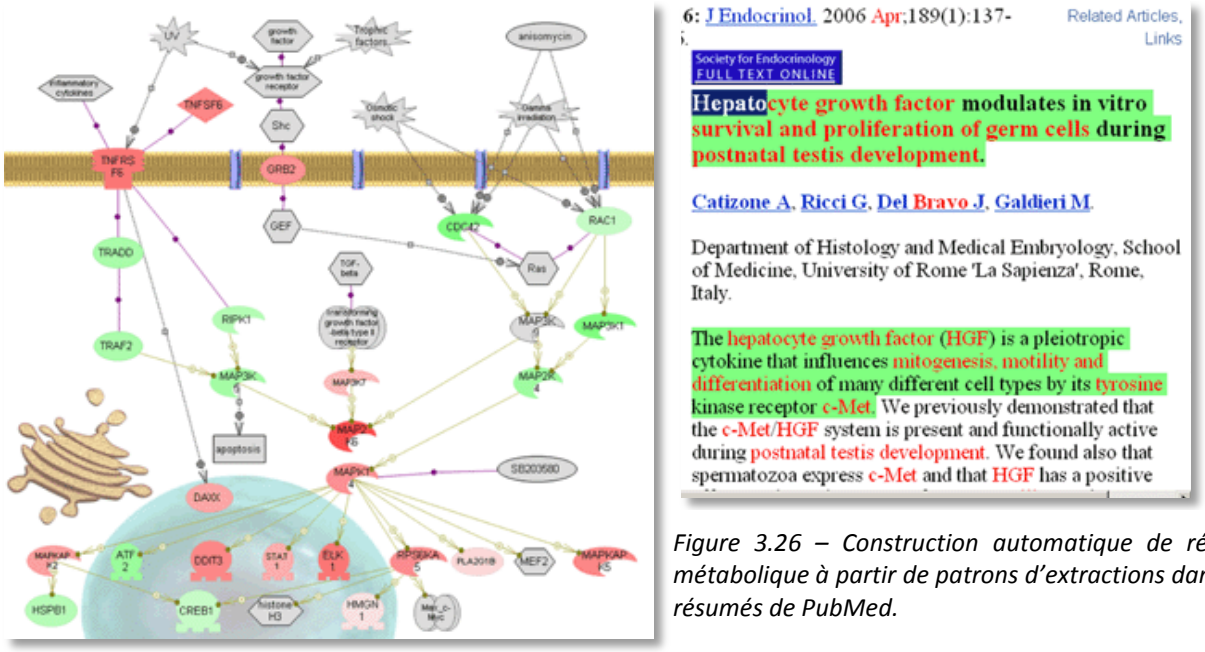


Figure 3.26 – Construction automatique de réseau métabolique à partir de patrons d'extractions dans les résumés de PubMed.

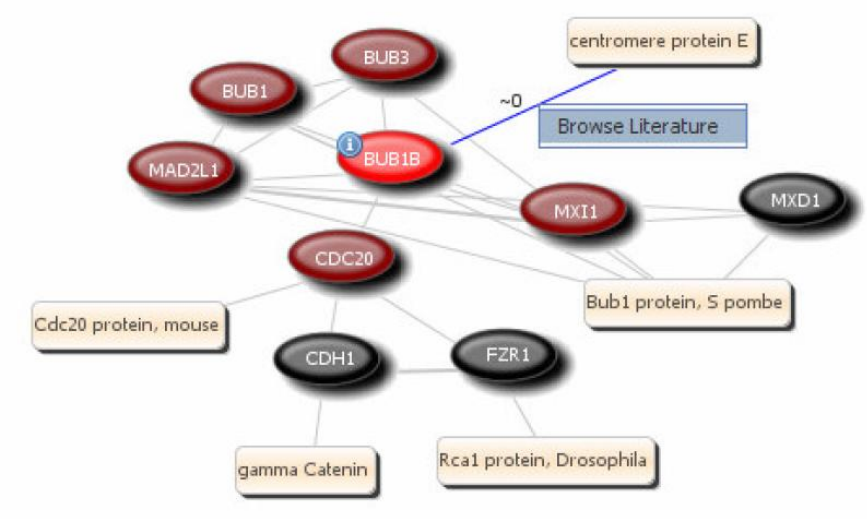


Figure 3.27 – PubGene, un outil d'analyse distributionnelle de texte et visualisant à l'aide de forces le réseau de cooccurrences obtenu. Les applications sont la recherche bibliographique et l'analyse d'un ensemble de gènes.

Une autre approche consiste à reconstruire ce graphe à partir de patrons syntaxiques. Les relations sont alors plus formelles et moins statistiques. L'avantage est en général de bénéficier d'une meilleure précision[1], mais la contrepartie est un plus faible rappel[2]. Le réseau obtenu reste cependant généralement particulièrement grand. Outre les réseaux de gènes, on trouve des réseaux d'interactions gènes-protéines ou protéines-protéines.

Les cartes de voies métaboliques sont des graphes reflétant les réactions enzymatiques présentes dans un métabolisme. KEGG en propose une approche pluri-espèces. Ces cartes sont construites manuellement à partir des connaissances de la littérature scientifique. Les données de KEGG ont ainsi été saisies par des experts. Des tentatives de reconstruction à partir de texte existent aussi, par exemple dans l'outil Pathway Studio (anciennement Pathway Assist) de Ariadne Genomics [Nikitin, Egorov et al. 2003] (figure 3.26). On retrouve enfin les projets de cellule virtuelle qui visent à simuler la dynamique d'une cellule et s'intéressent donc à une cartographie dynamique de la cellule (par exemples E-Cell et V-Cell).

[1] La précision est calculée par la formule suivante : $\frac{nombre\ d'éléments\ justes\ ou\ pertinents}{nombre\ total\ d'éléments\ dans\ la\ réponse}$

[2] Le rappel est calculé de la façon suivante : $\frac{nombre\ d'éléments\ justes\ dans\ la\ réponse}{nombre\ total\ d'éléments\ du\ corpus}$

Recherche d'information

Comme nous l'avons mentionné, cette recherche d'information représente un enjeu crucial pour les sciences du vivant ou plus généralement dans toutes les sciences. Les données de PubMed ont ainsi été exploitées par les outils Map.Net et TouchGraph (du nom de HubMed) par exemple. On trouve aussi des outils basés sur les statistiques distributionnelles les plus divers. Une autre approche est proposée par l'outil HealthCyberMap (figure 3.28). La recherche peut s'effectuer au travers d'une carte géographique, les pays permettant de choisir l'origine d'un document et donc la langue. Une originalité est de proposer une approche anatomique et de naviguer à travers un corps humain pour sélectionner une thématique.

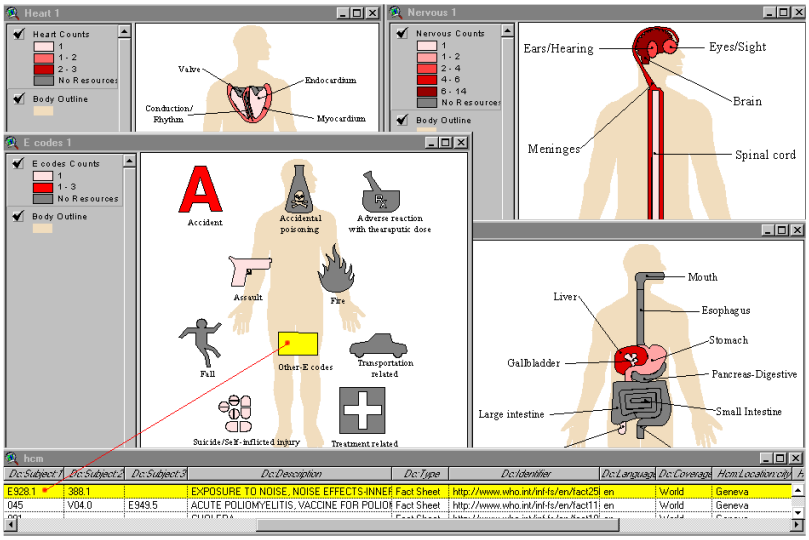


Figure 3.28 – HealthCyberMap, portail de recherche d'information médicale employant une carte anatomique pour déterminer le thème et une carte géographique pour déterminer la langue ou le pays d'origine.

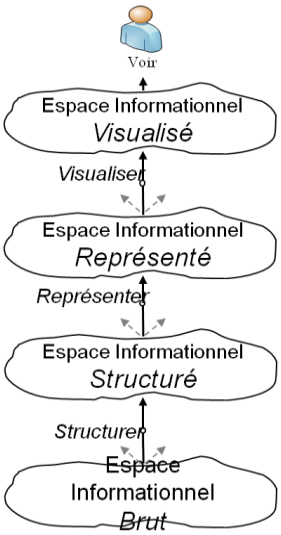
3.4.5 Cartographie & carte : un problème d'adaptation

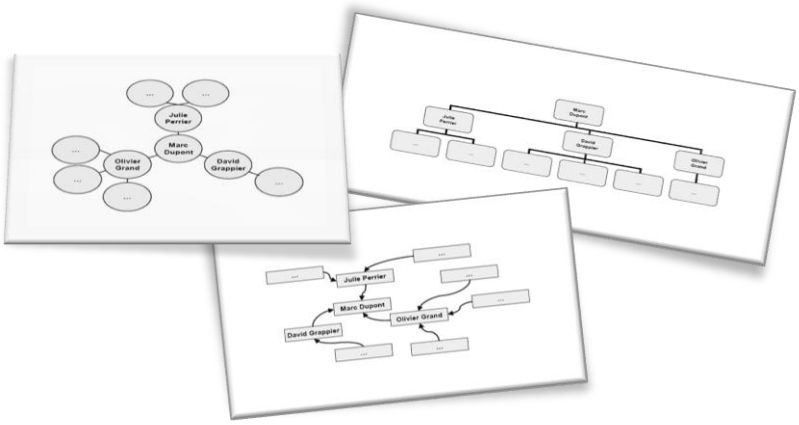
Nous avons eu l'occasion de découvrir de nombreuses cartes utilisées dans divers contextes. Un inventaire plus exhaustif a été maintenu à jour jusqu'au début de l'année 2004 par Martin Dodge, au travers du magazine *Mappa•Mundi* (disparu en 2001) et de son ouvrage *« Atlas of CyberSpaces » [Dodge and Kitchin 2001].* Il montre le nombre de travaux qui existent dans ce domaine.

Nous avons défini la cartographie comme la discipline et l'activité de construction d'une carte. A partir de données cartographiques, on souhaite générer plusieurs cartes adaptées à des contextes différents. C'est ce qui distingue notre approche et notre définition de la cartographie des nombreux exemples précédents : les données ou la sémantique ne sont le plus souvent pas réellement utilisées, et plus rarement encore réutilisées.

La communauté des géosciences respecte cependant cette définition. Nous avons par exemple mentionné le site *World Mapper* permettant de déformer le monde à volonté en fonction de différents critères (le revenu national, la population et la superficie). C'est aussi le travail proposé par Google Earth et Google Maps ou encore le GéoPortail de l'IGN en France : à partir de photos satellites ou aériennes de la surface de la terre, on propose de superposer des données diverses : noms des rues, des monuments historiques, résultats des élections, etc.

En dehors de la communauté des géosciences, peu d'approches informatiques se sont focalisées sur ces aspects. Deux projets existent à notre connaissance : la cartographie sémantique de Christophe Tricot et le projet MolAge [Crampes, Ranwez et al. 2006].





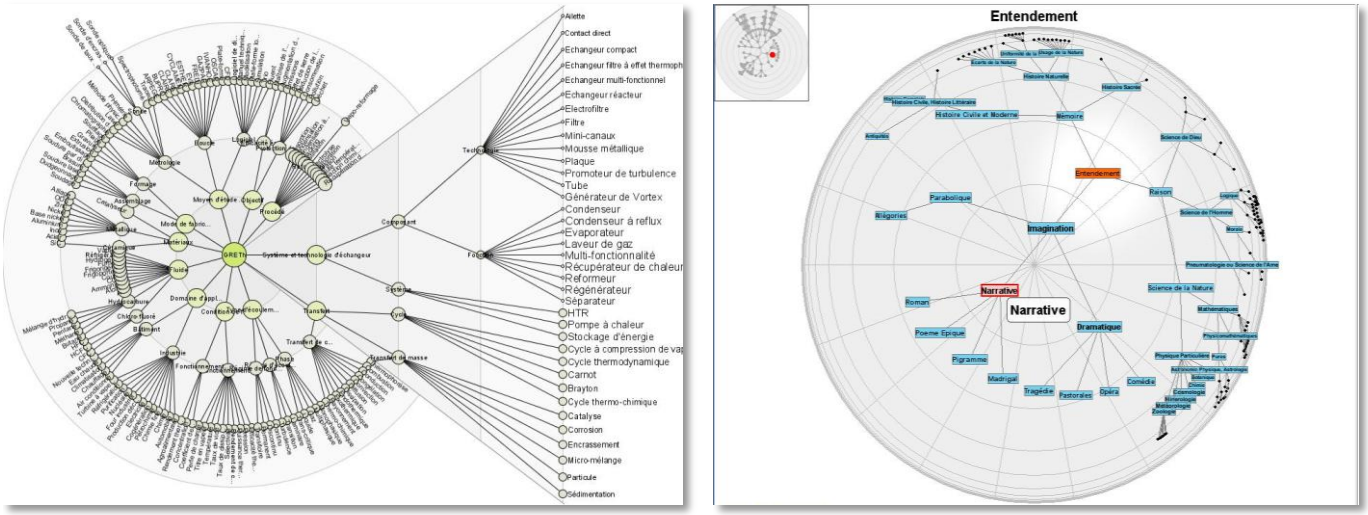
▲ *Figure 3.29 – Exemple de choix de représentation pour une structure donnée (ici un arbre), préalable au rendu graphique.*

◄ *Figure 3.30 – Découpage en quatre niveaux de la production d'une carte.*

Christophe Tricot s'intéresse dans sa thèse à la cartographie sémantique. Plus précisément il s'agit de cartographier la « sémantique d'un domaine », ce qu'il restreint à une hiérarchie de termes et concepts. Il décompose en quatre étapes la construction d'une carte suivant l'architecture préconisée dans [Chi and Riedl 1998] et [Mackinlay 1986] (figure 3.30):

- structuration d'un espace informationnel brut,
- représentation de cet espace informationnel structuré (figure 3.29),
- visualisation de la carte représentée,
- adaptation de la carte par l'interaction de l'utilisateur.

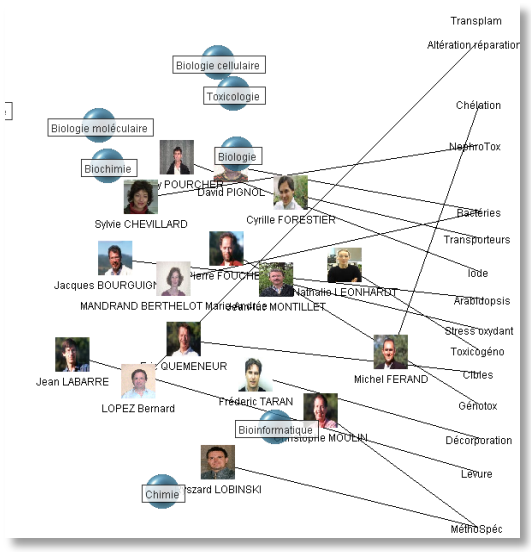
Concernant les deux premières étapes, il propose deux formalismes : SNDF permet de structurer les données sous forme d'un réseau sémantique, MDL est un dérivé d'XML qui permet de décrire les opérations de représentation de ces données (disposition dans la fenêtre, etc.). L'introduction d'une couche « représentation » est une évolution de modèle de Chi, précédemment introduite par M. Carpendale [Carpendale 1999]. Par la suite, un outil permet de générer une visualisation soit sous forme de « fish-eye polaire » (une représentation proche de la géométrie hyperbolique palliant la difficulté de manipulation de cette visualisation), ou par une disposition radiale avancée (figure 3.31).



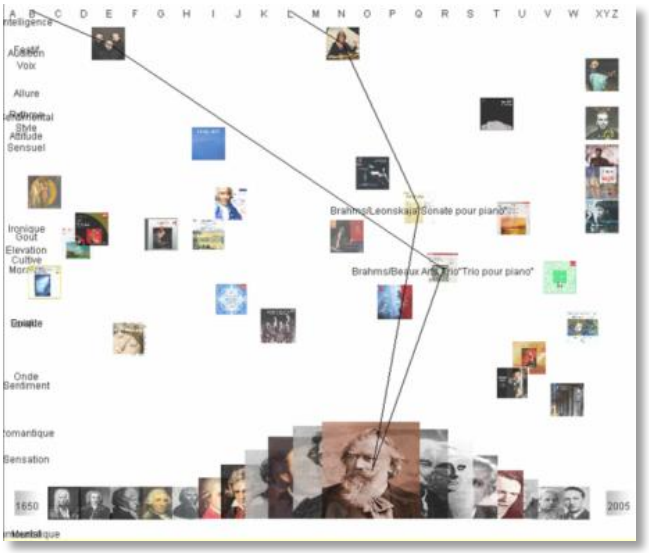
Visualisation radiale *« Fish-eye » polaire*

Figure 3.31 –Mise en œuvre de données cartographiques dans deux visualisations différentes, l'une étant plus accessible au débutant, l'autre adaptée à l'expert.

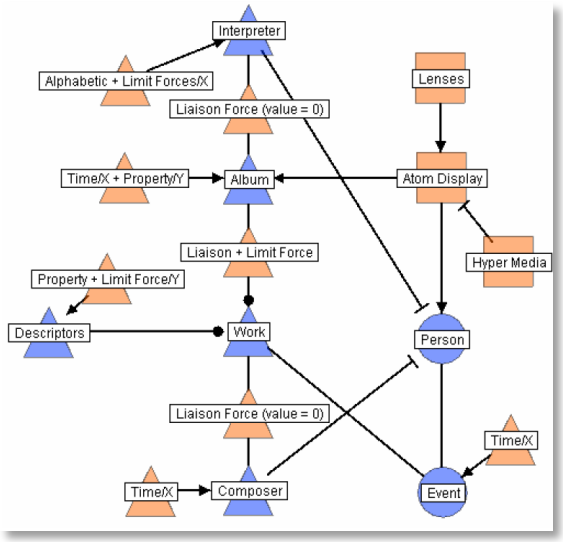
La seconde approche adoptée par [Crampes, Ranwez et al. 2006] représente dans un format propriétaire dérivé d'XML des données et des informations sur leur représentation et sur les lentilles mises en œuvre. Il est possible de combiner un graphe et des données multidimensionnelles. La visualisation est basée sur un modèle physique [Eades 1984]. L'environnement propose de multiples fonctionnalités et permet de composer de nombreuses vues différentes. L'illustration proposée (figure 3.32) montre quatre exemples d'application de l'environnement : la supervision du programme ToxNuc-E (↖) [Anoir, Penalva et al. 2005], l'exploration d'une base musicale (↗), la spécification d'une visualisation en alignant une ontologie du domaine avec un métamodèle de MolAge [Crampes, Villerd et al. 2006] (↙) et l'indexation de titre musicaux reprenant le principe des diagrammes d'Euler (↘).



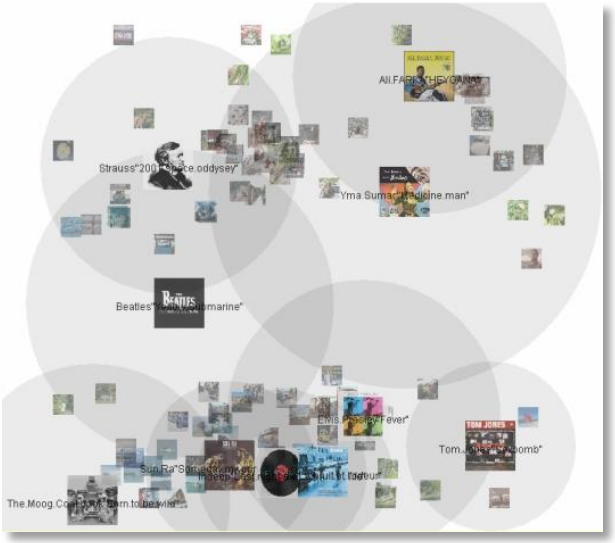
Supervision de projet



Exploration d'une CD-thèque



Spécification formelle d'une visualisation



Indexation de titres musicaux

Figure 3.32 – Captures de MolAge, un environnement de visualisation appliqué dans des contextes divers.

3.5 Synthèse

Le biologiste rencontre une difficulté face à une dispersion et à une hétérogénéité des données et des outils pour les manipuler. La réponse que nous proposons reprend le principe de la cartographie : nous souhaitons mettre en œuvre de multiples vues sur les données

adaptées au contexte et à la tâche à réaliser. Les données cartographiques des connaissances biologiques sont au centre de ces vues. Notre approche a ainsi l'originalité de s'intéresser conjointement à deux domaines : l'intégration de données et la visualisation de ces données.

Nous avons pour cela conçu un environnement appelé I^2DEE et présenté dans la partie suivante. Lors de la construction de cette solution, nous avons constamment dirigé nos réflexion afin d'offrir un environnement simple, souple, extensible, capable de répondre aux besoins diversifiés des utilisateurs et des développeurs.

L'informaticien peut ainsi être amené à exploiter I^2DEE pour des développer des procédures de fouille de données, de produire un portail en ligne, d'accéder à des services Web, de construire un outil riche adapté à un besoin spécifique ou encore ajouter une source dans le système. Il doit pouvoir réaliser tout cela facilement, et en un temps minimum.

L'utilisateur final au contraire a d'autres préoccupations : une équipe peut souhaiter disposer d'un entrepôt pour fouiller les données, et en être propriétaire pour les nettoyer ou les annoter. Un second cas d'utilisation est le portail en ligne qui est utilisé pour accéder à une information à tout instant, sans installation d'un outil quelconque. Enfin, le troisième cas est celui de l'application métier qui nécessite de croiser l'information avec plusieurs sources, et plusieurs applications métiers. On souhaite alors apporter une réponse concrète au problème « du nombre de fenêtres qui encombrent le bureau », en minimisant les accès nécessaires à des portails externes et en mettant à disposition des techniques de visualisation avancées.

Dans la suite de ce mémoire, nous présentons l'environnement I^2DEE. Après un description générale de l'architecture et du modèle qui apportent souplesse et extensibilité, nous détaillons les procédures d'intégration et de fouille de données mise en œuvre. La boîte à outil graphique est alors présentée, à l'issue de quoi nous détaillons comment à différents niveau de l'environnement, nous contribuons à l'adaptabilité du système. Enfin, nous concluons en présentant l'application d'I^2DEE à deux problèmes spécifiques et distincts : la conception d'une ontologie et l'analyse de données d'expression issues de puces à ADN et provenant de plusieurs jeux de données distincts.

Partie 2

L'environnement I^2DEE : méthodologie, mise en œuvre et résultats

CHAPITRE 4

Présentation générale d'I²DEE

« How can experimental protocols, descriptions of model systems, statistical criteria for data acceptability, and many other critical elements be effectively communicated between technology silos? »

TED SLATER

4.1 Introduction

Notre contribution pour répondre à la problématique posée dans la première partie consiste à reprendre une approche traditionnelle issue de la cartographie géographique : un système d'information géographique contient des données qui sont contextualisées par rapport à un besoin sous la forme d'une carte. Cette carte est visuelle, intuitive, adaptée à un besoin spécifique. Dans le contexte biomédical, l'application de cette approche nous amène à proposer un entrepôt de données biologiques et une boîte à outils de visualisation permettant la conception rapide et simple d'interfaces utilisateurs adaptées à un besoin spécifique. Notre architecture se veut avant tout souple et ouverte afin de prendre en compte les problématiques sociologiques mises en évidence par L. Stein [Stein 2003].

I²DEE[1] est l'environnement que nous avons implémenté en fonction de cette architecture. Notre approche vise à respecter un compromis entre expressivité et extensibilité. Pour cela, nous proposons une approche simple à base de graphe. Le modèle peut être perçu comme un métamodèle, et positionné à l'intersection des systèmes d'intégration et des navigateurs (systèmes à base de liens et chemins) : il propose une représentation navigationnelle de l'information tout en conservant la finesse de représentation d'une base de données et le langage de requête du système d'intégration. La première partie de ce chapitre décrit ce choix de modélisation. La seconde partie montre comment en s'intégrant plus globalement dans l'architecture d'I²DEE, ce modèle permet de répondre de façon unifiée à la plupart des besoins qui motivaient jusqu'ici des approches différentes de l'intégration de données. Les principales composantes sont détaillées dans les chapitres qui suivent (procédure d'intégration, boîte à outils graphique, mécanisme d'adaptabilité).

4.2 Modèles des données

4.2.1 Vers une modèle de graphe

Un modèle extensible et simple

Les démarches existantes des systèmes d'intégration consistent à proposer un schéma basé sur un modèle riche et une description profonde du domaine. Actuellement de nombreux systèmes reposent sur plusieurs centaines de tables, les plus simples en contiennent de 40 à 80. Les démarches consistant à proposer un modèle plus complexe pour étendre un système ne nous ont pas convaincus. Nous adoptons un autre point de vue : concevoir un système d'information extensible consiste à lui permettre de s'étendre sans remettre en cause sa structure, son schéma, ou son architecture. Un système extensible possède avant tout un schéma simple et expressif. Un schéma simple procure d'autres avantages :

- la réduction de la charge de travail pour le développeur et le système,
- la facilité de compréhension par l'utilisateur, éventuellement simplifier la spécification des requêtes.

Lorsque les schémas sont complexes, leur intégration n'en est que plus difficile. Prenons l'hypothèse d'un premier modèle stockant des séquences nucléotidiques. Les données doivent être intégrées dans un second modèle plus complexe, faisant une distinction entre ADN, ARN messager et ARN interférent. La distinction entre ADN et ARN peut être réalisée automatiquement en fonction de la présence d'uracile ou de thymine dans la séquence. Cela illustre qu'une procédure d'intégration ne peut être totalement automatisée et doit être définie

[1] *an Integrated and Interactive Data Exploration Environment*

par un expert à l'aide d'un langage suffisamment expressif pour programmer des procédures complexes. Le second problème est de savoir s'il s'agit d'un ARN messager ou interférent. Deux cas se posent : soit le modèle de destination propose un mécanisme de généralisation pour l'ADN, soit il est impossible d'intégrer les données sans complément d'information.

Le cadre unificateur que nous proposons doit posséder un schéma simple et extensible. C'est la stratégie employée par les systèmes à base de liens et c'est l'une des raisons de leur succès : la simplicité d'ajout d'une source et la simplicité d'utilisation. Par ailleurs, au travers d'outils comme BioGuide et Getz (SRS), on constate que la simplicité n'interdit pas l'existence de langages d'interrogation complets, fonctionnels et performants. La limite principale des navigateurs provient du niveau de granularité de la représentation. SRS et Entrez manipulent des pages, des documents, BioGuide et BioNavigation manipulent des types, mais ne permettent pas de manipuler leurs instances. Notre positionnement est de considérer qu'il faut représenter les ensembles mais aussi les instances, à un niveau de granularité élémentaire selon la première forme normale de Codd.

Un graphe

On peut classer un grand nombre de structures de données existantes en deux catégories : les approches à base de graphe et les approches multidimensionnelles. Pour être extensible, le modèle que nous adoptons doit permettre de concilier ces deux approches le mieux possible, en prenant en compte les problématiques de visualisation. Il existe plusieurs méthodes permettant d'insérer un graphe dans un espace multidimensionnel [Gaume 2004; Vert 2004]. La position d'un élément du graphe dans chaque dimension n'a alors généralement pas de sens. Le problème est identique lorsque l'on souhaite intégrer deux espaces multidimensionnels ne partageant pas les mêmes dimensions : seuls les distances et les voisinages à un niveau global sont significatives. De plus, lorsque l'on souhaite visualiser un espace multidimensionnel, on est généralement amené à réduire le nombre de dimensions entre un et trois. Les dimensions ne sont alors plus significatives ; parfois même la dimension, à un niveau individuel, n'a pas de sens dans l'espace original. Par exemple, dans les expériences sur puces à ADN, l'expression d'un gène à une heure donnée n'est pas exploitable : on exploite la similarité de l'expression des gènes sur l'ensemble de l'expérience.

A partir de la matrice de distances, il est possible de générer un graphe complet et de le projeter. Chaque arête a une longueur définie en fonction de la distance dans l'espace original. Cette solution s'avère efficace à l'usage[1], mais nécessite une grande capacité de mémoire[2]. Cela procure différents avantages :

- on représente les relations entre éléments,
- les algorithmes et outils issus de la théorie des graphes sont très nombreux,
- cette structure est souvent considérée comme plus simple à appréhender pour l'utilisateur final que l'espace multidimensionnel.

L'inconvénient principal est la disparition de la signification des dimensions. Cependant, comme nous l'avons mentionné, cela se produit généralement lorsque l'on souhaite intégrer des espaces multidimensionnels distincts et les visualiser. Cependant, si la sémantique de la dimension revêt un intérêt crucial, il est possible de représenter chaque vecteur-dimension comme un nœud du graphe, et associer une distance entre la dimension et chaque élément. On peut finalement les projeter simultanément dans le plan [Crampes, Villerd et al. 2006]. Dans de nombreux cas cependant, visualiser les dimensions et leur orientation n'est pas souhaitable. Dans les approches saltoniennes de la recherche d'information, une dimension n'a pas réellement de sens ; seule la distance qui émerge dans l'espace et le voisinage est significative à un niveau global. Dans le contexte biologique, il en va de même avec les données d'expression de

[1] Nous détaillons une évaluation de la qualité dans le chapitre 6.

[2] Soit n le nombre de vecteurs et d le nombre de dimensions, la taille de l'espace vectoriel est $n \times d$, la taille de la matrice de distance est n^2.

gènes par exemple. Lorsque l'on étudie l'expression d'un gène de *Plasmodium Falciparum* pendant 48h, l'expression d'un gène à un instant donné est difficilement exploitable individuellement. Nous avons choisi de représenter nos données suivant un modèle simple de graphe, en représentant des types abstraits atomiques et des données relatives aux instances de ces types.

Le modèle relationnel propose des mécanismes performants grâce à l'algèbre sur laquelle il repose et à ses opérateurs ensemblistes (union, intersection, etc.) et spécifiques (sélection, projection, etc.). Cependant, les mécanismes offerts par le langage de requêtes ne sont pas suffisants : il n'y a pas de mécanismes d'itération, d'outil de visualisation, etc. Nous devons donc recourir à un ou plusieurs langages externes afin de satisfaire nos besoins (PL/SQL, Java, etc.). Nous proposons deux modèles : un modèle relationnel adapté au stockage non redondant des données, et un modèle orienté objet qui permet la manipulation des données et leur contrôle au travers d'une interface métier.

4.2.2 Modèle relationnel

Le modèle conceptuel proposé est décrit dans la figure 4.1. Le graphe est modélisé par un ensemble de nœuds (`Node`) et d'arêtes (`Edge`), une arête étant associée à un nœud source et un nœud destination. Afin de proposer une représentation plus riche, nous proposons de décrire ces nœuds et ces arêtes à l'aide de propriétés. `Property` est l'ensemble des types de propriétés existants. Ces types peuvent éventuellement être hiérarchisés. Un type prend une valeur pour un nœud ou une arête donnée.

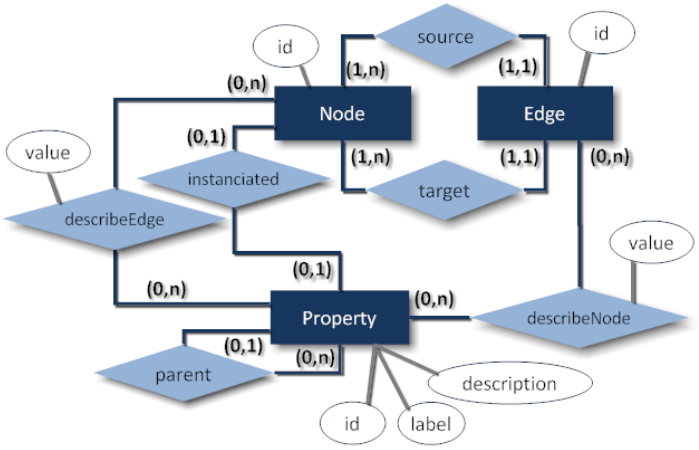


Figure 4.1 – Modèle conceptuel de données

La notion de propriété est très générale : le type d'un nœud ou d'une arête, sa provenance, le résumé d'un article, le nom d'un auteur. La hiérarchisation des types n'est pas une obligation, mais s'avère utile, par exemple, pour gérer la notion de date (de soumission, de publication, de révision, etc.), de titre (original ou anglais), de résumé (original ou anglais), de nom, de séquences (ADN, ARN, ADNc, protéine), etc. Ce schéma est simple mais riche d'expression. On peut le considérer comme un métamodèle. Des approches similaires existent déjà avec RDF, par exemple, qui est utilisé par des langages d'ontologies, Dublin Core, etc. Les nœuds et arêtes sont des données, l'entité Property regroupe l'ensemble des métadonnées.

Un problème subsiste : comment représenter la propriété de « taxonomie ». Par exemple, comment représenter qu'une annotation (arête entre un gène et un concept) est prouvée pour un organisme donné. L'organisme est ici une propriété relative à l'arête. Mais la taxonomie est aussi un nœud, un concept de nos ontologies. Pour répondre à ce besoin, nous avons ajouté une association entre le type de propriété et un nœud qui la décrit. Le modèle relationnel correspondant à ce modèle conceptuel est représenté dans la figure 4.2. Nous y dissocions alors les métadonnées relatives à des nœuds et celles relatives à des arêtes.

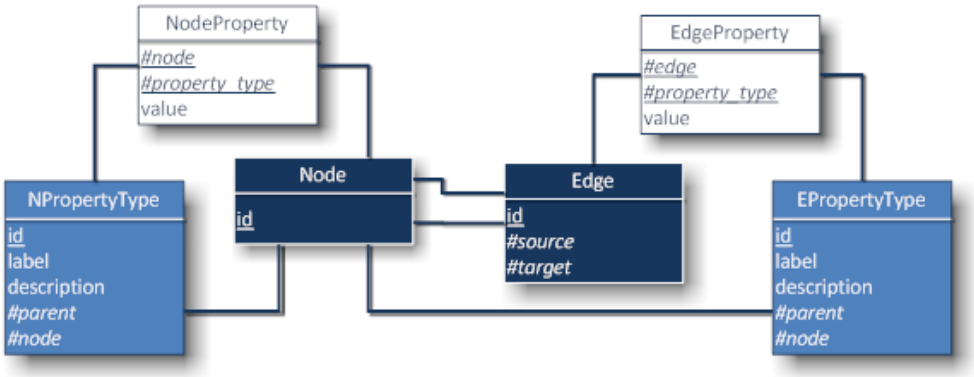


Figure 4.2 – Modèle relationnel de données – Les clés primaires sont soulignées, les clés étrangères sont préfixées de « # » et en italique. Les entités du MCD sont bleu foncé, les tables correspondant à des métadonnées bleu clair, et les tables générées par des associations « many-to-many » sont blanches. Cette sémantique des couleurs est conservée dans le schéma physique qui suit. On peut ainsi considérer que les métadonnées sont en bleu clair, les données sont alors contenues par les tables blanches et bleu foncé.

Le schéma physique que nous proposons diffère fortement du schéma relationnel précédent. D'une part, nous introduisons une distinction entre propriétés, au sein des métadonnées. Dans les exemples de type de propriétés que nous avons apportés précédemment, nous avons dissocié deux grandes catégories. La première regroupe des propriétés portant sur l'ensemble des nœuds et arêtes : tous ont un type et une source d'origine par exemple. La seconde est au contraire un ensemble de propriétés qui sont spécifiques à un type ou à une source donnée. Par exemple, une séquence ne concerne que les gènes, ARN et protéines, le titre vernaculaire n'est renseigné que pour une faible partie des documents de PubMed, et les définitions ne sont disponibles que pour un petit sous-ensemble des concepts d'UMLS.

Nous avons ainsi identifié quatre propriétés principales dites « statiques » : il n'est pas possible de faire évoluer leur nombre au cours de l'exécution. Ces propriétés sont (figure 4.3) :

- le type et la source d'un nœud ou d'une arête,
- et la preuve et la taxonomie qui sont relatives à une arête uniquement. Une arête ne prend pas systématiquement de valeur pour ces deux dernières propriétés.

Entité	Propriété	Description	Exemples
Noeud	**Type**	Type du nœud	*Element biologique (gène, protéine), document (article, synthèse), acteur (auteur, relecteur), date (soumission, publication), concept (concept ou type sémantique d'UMLS), …*
	Source	Provenance (BD) du nœud	*Genbank, UMLS, Gene, Ontology, PubMed, PlasmoD, …*
Arête	**Type**	Type d'arête	*Relation sémantique, relation statistique (cooccurrence, collocation, …), auteur/document, gène/protéine, réaction métabolique, médicament/affection, …*
	Source	Provenance de l'arête	*Genbank, UMLS, Gene, Ontology, PubMed, PlasmoD, …*
	Preuve	Méthode ayant permis d'établir cette arête.	*Annotation manuelle ou automatique (phylogénie, etc.), regrouppement flou ou classification hiérarchique, …*
	Taxonomie	Organisme pour lequel l'arête est vérifiée.	*Relation conceptuelle (non défini), annotation d'un gène (Homo sapiens, mus musculus, ….), …*

Figure 4.3 – Types de propriétés des nœuds et arêtes

Le modèle physique présenté dans la figure 4.4 implémente ce choix : six tables spécifiques contiennent les métadonnées des propriétés globales à l'ensemble de l'entrepôt (`N_Type`, `N_Source`, `E_Type`, `E_Source`, `E_Evidence`, `E_Taxonomy`). Toutes les autres propriétés sont appelées « champs » (*« fields »*). On peut ajouter un nouveau type de champ à tout moment et sans intervenir sur le schéma des données. Les types de champs sont regroupés dans les tables `N_Fields` et `E_Fields`, et les valeurs prises par les nœuds et arêtes sont stockées dans des tables intermédiaires.

La combinaison des champs et propriétés statiques est importante : les propriétés statiques apportent structure et performance. Les requêtes ne nécessitent pas de jointure pour appliquer un critère de sélection ou de tri sur ces attributs. Cela améliore les performances du système et permet de spécifier des contraintes d'intégrité référencielle permettant de garantir une consistance minimale des données dans l'entrepôt. Les champs apportent au contraire de la souplesse : leur inventaire peut évoluer à tout moment et sans limite. Si, par exemple, on met à disposition de l'utilisateur un nouvel outil lui permettant de saisir des commentaires personnels, il est possible de créer dans l'entrepôt un nouveau type d'attribut « COMMENT_MySoft ». Cette extensibilité se fait au détriment de la structuration : il devient très coûteux de vouloir contrôler au sein du SGBDR qu'on ne rajoute pas une séquence à un auteur, un titre à un concept, etc. Cela nécessite en effet de spécifier des procédures en PL/SQL et de les associer à un déclencheur (trigger).

Les champs permettent aussi une économie d'espace : certains champs étant rarement complétés (par exemple les définitions de concepts, les titres vernaculaires, …), on n'alloue pas inutilement de la mémoire pour l'ensemble des éléments de l'entrepôt.

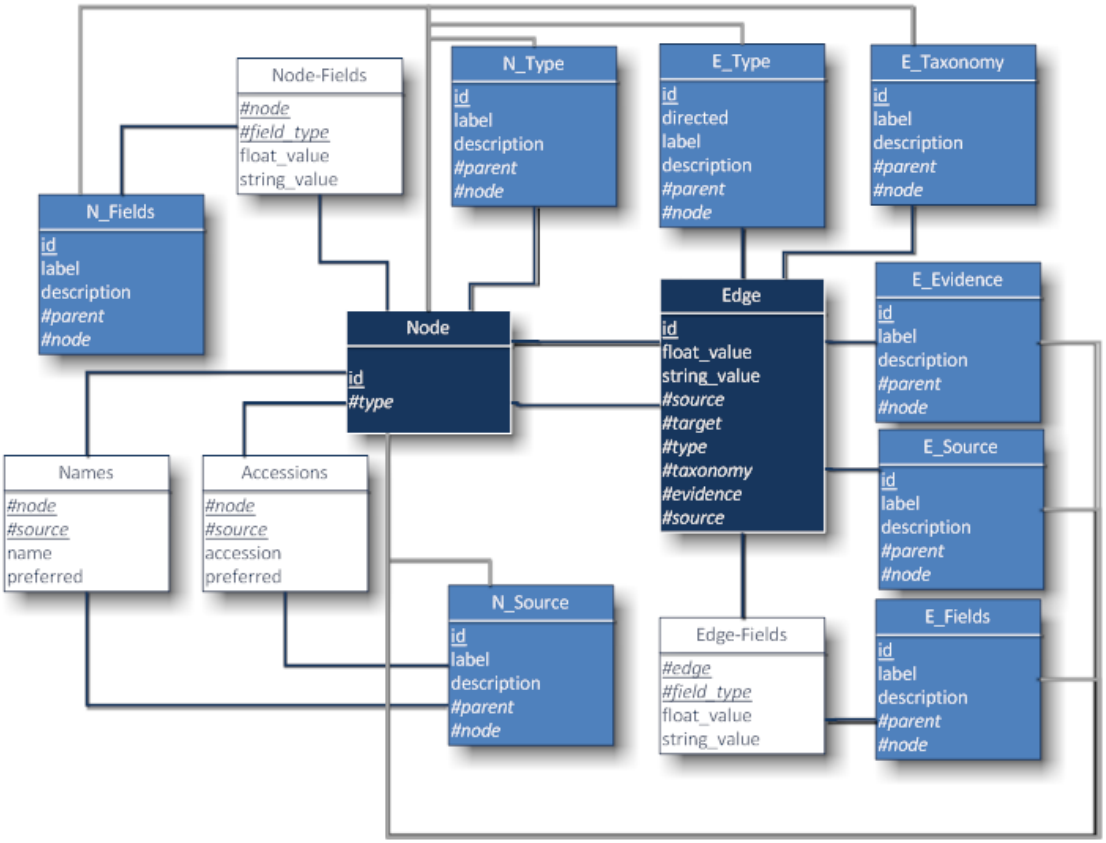


Figure 4.4 – Modèle physique de données – Les clés primaires sont soulignées, les clés étrangères sont préfixées de « # » et en italique. Le code de couleurs est le même que pour la figure 4.2.

Une seconde évolution importante dans le modèle physique concerne la gestion des sources. La source pour une arête est gérée comme un type : deux relations entre des mêmes nœuds et contenant la même information mais provenant de deux sources différentes produisent deux instances. Concernant les nœuds, la modélisation diffère : une instance peut provenir de

plusieurs sources et pour chaque source posséder plusieurs noms (termes et entités nommées) ou numéros d'accession. Pour un nœud et une source donnés, un terme et un numéro d'accession peuvent être préférables.

Nous venons de présenter les schémas de données de l'entrepôt qui s'apparentent à un métamodèle. Ce schéma est particulièrement simple puisqu'à partir de trois entités dans le MCD, nous aboutissons à un ensemble de 14 tables. Bien que minimaliste, ce schéma permet de représenter un contenu riche et d'accéder aux données et métadonnées. Les propriétés statiques apportent une structuration essentielle et des améliorations sensibles des performances. Les champs apportent souplesse et extensibilité. Le schéma est facile à appréhender pour le développeur et permet de réduire le coût d'implémentation des procédures d'intégration. Contrairement à BioGuide et BioNavigation, il représente les instances des types et des sources. Contrairement à SRS, il permet de représenter des données atomiques.

4.2.3 Modèle objet

Pour des raisons de lisibilité des schémas, nous présentons le modèle d'une façon incrémentale et partitionnée. Nous avons employé le formalisme UML (diagrammes de classes) présenté en annexe (page 237). Toujours pour des raisons de lisibilité, nous avons omis certains détails (attributs, méthodes, accessibilité, certaines associations, etc.).
Les schémas sont par ailleurs conçus en couleur. Si les schémas sont parfaitement lisibles en noir et blanc, la couleur permet une lecture plus confortable. La signification des couleurs est la suivante : les relations et associations sont en noir, les interfaces en bleu, les classes abstraites en vert, et les classes concrètes (instanciables) en jaune. Les noms des classes abstraites sont préfixés par « Abstract », les implémentations par défaut sont préfixées par « Default ».
Enfin, pour respecter les pratiques courantes de modélisation, nous avons choisi de définir un grand nombre d'interfaces qui ont pour objectif de permettre de structurer hiérarchiquement notre modèle, mettre en valeur les rôles principaux et favoriser l'extensibilité. Ainsi, plusieurs interfaces ne possèdent qu'une seule implémentation et sont vides de code ou presque.

Une présentation détaillée du formalisme des diagrammes de classe d'UML est proposée dans l'annexe B.2 (page 271).

Le patron général de graphe que nous utilisons (figure 4.5) est proche du modèle conceptuel présenté précédemment (figure 4.1, page 121). Il distingue les propriétés statiques des champs comme cela est matérialisé dans le modèle physique de données.

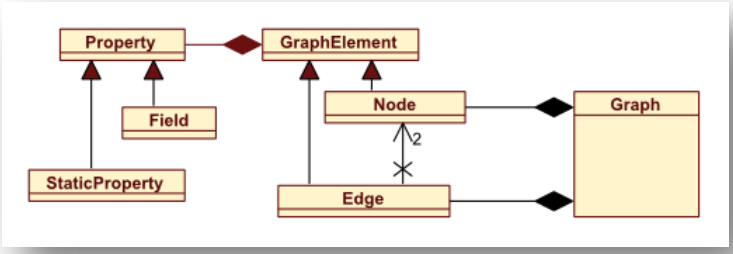


Figure 4.5 – Modèle simplifié du graphe

Les schémas qui suivent prennent en compte les pratiques courantes de la programmation orientée objet. Le modèle est globalement structuré hiérarchiquement à l'aide d'interfaces. Le code de ces interfaces est souvent limité voire inexistant : leur rôle est de laisser la possibilité d'étendre le schéma et de proposer des implémentations alternatives. Le schéma complet est dessiné dans la figure 4.6. Dans la hiérarchie, on distingue dans un premier temps les éléments de données des métadonnées. Au sein des métadonnées, on distingue dans un second temps les

structures contenant l'ensemble des propriétés décrites des instances de ces propriétés. On retrouve dans les deux cas la distinction entre propriété statique et champs. Cette distinction n'est pas nécessaire techniquement, mais permet d'obliger l'utilisateur-développeur à appréhender cette notion et favorise les bonnes pratiques. Dans la seconde partie du schéma, on retrouve toutes les données de l'entrepôt : nœuds, arêtes, références vers les propriétés statiques, valeurs prises pour les sources et champs.

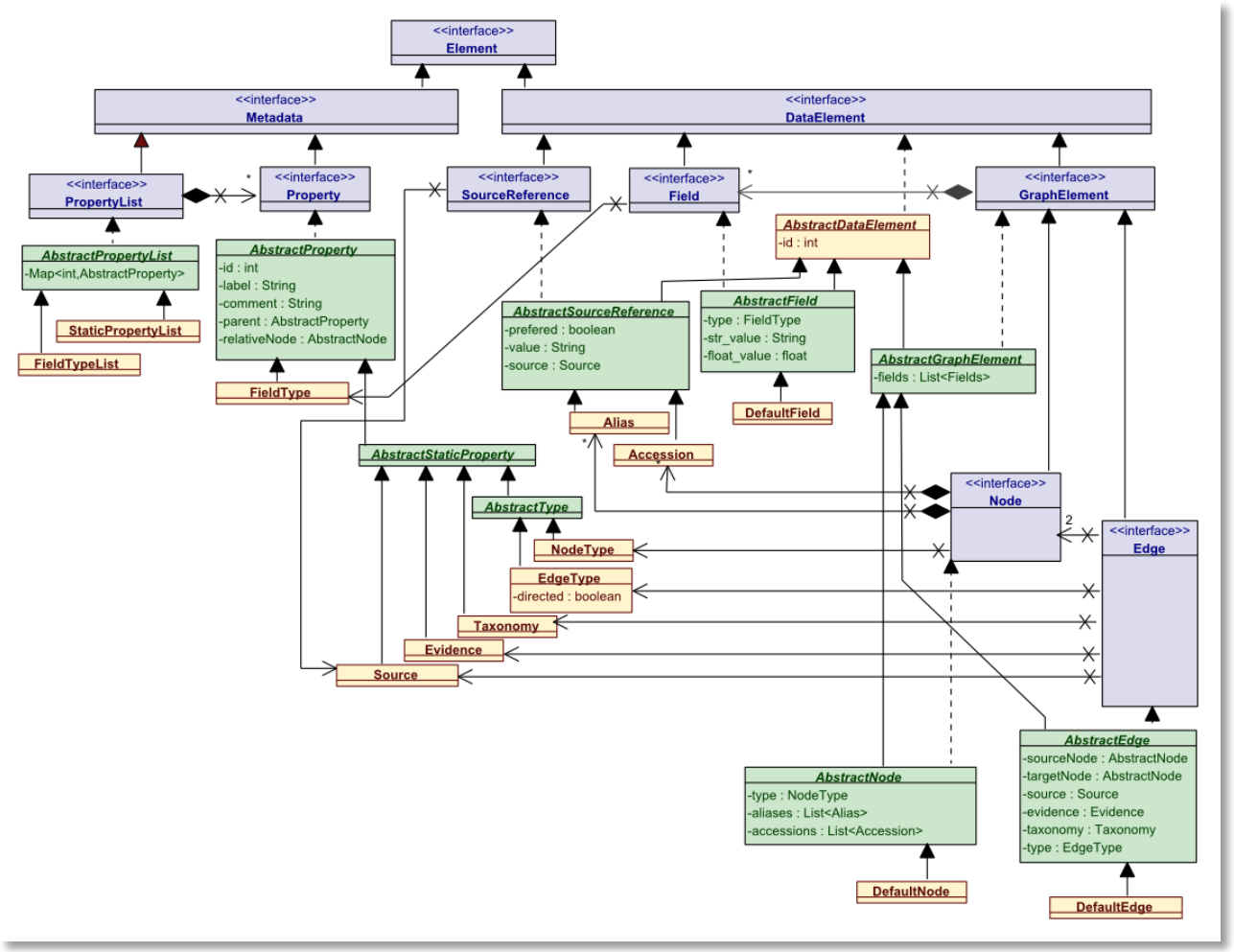


Figure 4.6 – Modèle objet de l'entrepôt de données utilisé dans l'API Java.

4.3 Architecture générale

Nous venons de présenter notre modèle de graphe et ses qualités : souplesse, extensibilité, légèreté et expressivité. Jusqu'ici les approches se focalisaient sur l'utilisateur ou le développeur, les besoins de ces deux acteurs apparaissant comme contradictoire. Nous débutons cette section en montrant qu'il est possible de réconcilier les approches existantes au travers d'I²DEE, qui se positionne comme un cadre unificateur. Dans un second temps, cette section explique comment s'appliquent les principes issus des sciences de l'information géographique dans I²DEE pour répondre aux besoins de l'utilisateur final. Nous illustrerons nos propos sous forme de scénarios et montrerons que la carte est un réel *support* à la *capitalisation* et au *partage* des connaissances. Enfin, l'architecture logicielle implémentée est présentée.

4.3.1 Polyvalence

Nous avons discuté dans le chapitre 3 du problème sociologique et du problème technique de l'intégration de données. En particulier, nous pensons que la technique n'a pas suffisamment pris en compte la dimension sociologique. Jusqu'ici, les besoins du développeur et de l'utilisateur sont apparus comme divergeant. Ceci n'est pas inéluctable, et nous souhaitons montrer comment un modèle extensible, notre métamodèle, permet de couvrir l'essentiel des besoins de l'utilisateur et du développeur (figure 4.7). Nous sommes en effet convaincus que des progrès tangibles ne peuvent être réalisés qu'en proposant une réponse commune satisfaisant les besoins de l'utilisateur et du développeur.

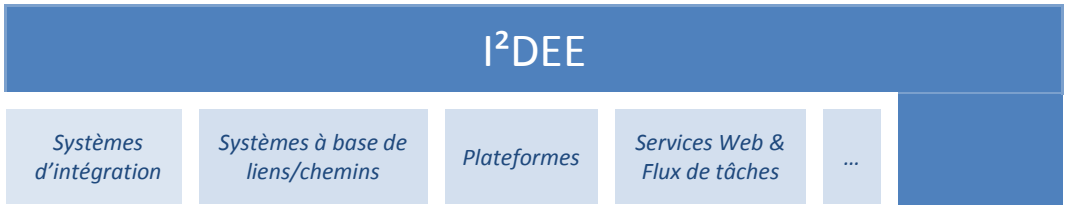


Figure 4.7 – (Rappel de la figure 3.8). Notre métamodèle, de par son extensibilité, sa légèreté et sa souplesse, se veut polyvalent. Il se positionne comme un cadre unificateur des approches existantes.

Système d'intégration

Les données de l'entrepôt sont persistantes dans un SGBDR et sont donc interrogeables au travers du langage SQL. Cependant, le modèle et les opérations sont éloignés des modèles classiques utilisés dans la plupart des systèmes d'intégration. La solution consiste à intercaler une interface métier basée sur des vues et des procédures (PL/SQL). Le modèle n'est pas optimisé pour ce modèle, mais on peut pallier le manque de performances en faisant le choix de vues matérialisées ; les principaux SGBDR le permettent. Le système distant peut alors reproduire le schéma des vues et matérialiser les données (approche entrepôt) ou interroger à distance (approche virtuelle). La seconde limite liée au schéma est qu'il est insuffisamment structurant. Si cela s'avère nécessaire, il est cependant possible de définir des procédures et de leur associer des déclencheurs (*triggers*). A nouveau, cela peut produire une dégradation des performances. Du point de vue du développement, il est moins coûteux de définir des vues et des contraintes que d'implémenter les procédures d'intégration et de fouilles existantes.

Systèmes à base de liens et chemins

Les environnements BioGuide et BioNavigation sont indépendants d'une plateforme. BioGuide a montré sa capacité d'adaptation rapide dans un environnement de workflow et pour interroger SRS *via* le langage Getz. La structure de graphe de SRS et de son langage est assez proche de notre environnement et nous n'avons entrevu *a priori* aucune limite théorique qui interdise à BioGuide de s'adapter à I²DEE.

Portail en ligne

Le portail généraliste en ligne est indispensable pour la communauté. On peut alors choisir plusieurs directions. La première consiste à réutiliser des interfaces fournies avec des systèmes d'intégration comme GUS ou AceDB (*alias* Wormbase). Ceci s'avère possible en mettant en œuvre les mécanismes de vue abordés dans le paragraphe dédié aux systèmes d'intégration. La seconde direction consiste à proposer un nouvel environnement générique. L'avantage des navigateurs existants est de proposer des documents (fichiers) construits vis-à-vis de thématiques précises (gène, protéine, etc.). La mise en page et l'ordre des données sont alors définis vis-à-vis de besoins récurrents dans la communauté. Dans le contexte de notre approche : comment interroger l'outil afin de spécifier l'élément de donnée au cœur du problème de

l'utilisateur ? Quelle information présenter à l'utilisateur ? Avec quelle profondeur dans le graphe ? Suivant quelle ordre et disposition ?

Une solution consiste à définir des patrons de recherche et des feuilles de styles personnalisables. SRS et BioGuide par exemple proposent des mécanismes de sélection et de préférence comparables. Cette solution est moins coûteuse à développer qu'un système complet et elle s'avère générique et réutilisable. De plus, elle permet à l'utilisateur de personnaliser la vue des données et de l'homogénéiser entre différents portails implantés.

Service Web

Il est possible de proposer des services Web pour l'accès aux données. Toutes les entités sont décrites à un niveau atomique et par des métadonnées. Il est possible à partir de là de générer automatiquement des scripts de déploiement de services ou des services génériques. Actuellement, nous n'avons pas mené de réflexion réelle sur la description du typage en ne distinguant que le contenu numérique du contenu textuel. Les valeurs numériques se stockent, se comparent et se calculent de façon performante. Une chaîne de caractères permet de coder de nombreuses informations. Ce typage n'est pas suffisamment fin pour BioMoby, par exemple, et il faudrait intercaler une couche de typage plus fine.

Migration de plateformes existantes

Les diverses plateformes existantes sont particulièrement hétérogènes, plus ou moins ouvertes et extensibles. Il est donc difficile d'émettre des hypothèses. Nous pouvons affirmer que notre modèle est simple à appréhender pour le développeur, accessible via des opérateurs métiers, des vues, un langage de requête, éventuellement des services Web, etc. I^2DEE ne pose aucun frein à l'extension de plateformes existantes, bien au contraire, sa simplicité et son ouverture ôtent les principaux obstacles qui se dressent devant le développeur.

Conception de nouveaux outils

C'est principalement dans cette optique que nous avons développé I^2DEE : permettre au développeur de concevoir rapidement des applications intégrant des données et les visualisant efficacement. La généralisation de notre approche permettrait l'homogénéisation de l'accès et de la manipulation des données pour l'utilisateur. La section suivante présente l'architecture de l'environnement d'un point de vue fonctionnel pour l'utilisateur.

4.3.2 Principe général d'utilisation

Présentation général

L'architecture que nous adoptons reprend un principe traditionnel en cartographie géographique : à partir d'un système d'information global (l'entrepôt) contenant de nombreuses données géographiques, de multiples cartes sont produites dans différents contextes : cartes routières, maritimes, pour les randonneurs, au sein de terminaux mobiles, sous une forme « papier », à différentes échelles, etc.

L'architecture d'I^2DEE illustrée dans la figure 4.8 respecte ce principe : un entrepôt de données est construit en intégrant de multiples ressources. Au travers de l'application cliente, l'utilisateur demande à l'entrepôt l'extraction d'une carte contextualisée. Un contexte peut par exemple être lié à un organisme, une affection, la génomique, etc. Cette extraction relève d'une exportation : il n'y a pas de synchronisation permanente entre l'entrepôt et la carte. La carte peut être un simple fichier XML ou SQL.

La carte est le support pour le travail collaboratif du chercheur. Elle l'accompagne dans les diverses tâches qu'il doit accomplir. Une carte peut être partagée par plusieurs utilisateurs et plusieurs applications. Elle vise à faciliter la capitalisation des connaissances de l'équipe et l'exploitation de ces connaissances collectives au sein des applications métiers.

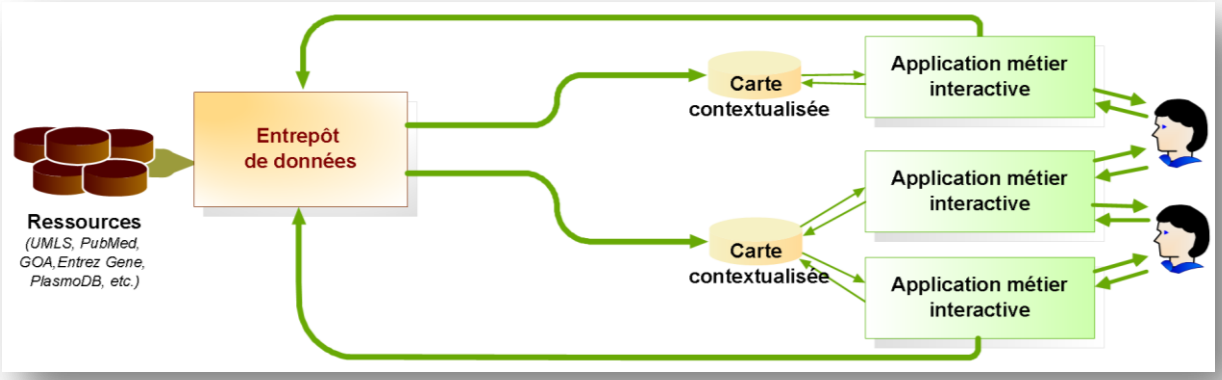


Figure 4.8 – Organisation générale : l'entrepôt permet la génération de cartes contextualisées qui peuvent par la suite être mises à jour si l'utilisateur le souhaite. Ces cartes ne sont cependant pas synchronisées de façon systématique avec l'entrepôt.

Dans la suite de cette section, un scénario hypothétique montre en quoi la carte est un support partagé, un outil collaboratif. Nous terminons par une présentation plus détaillée de l'architecture à l'aide d'un diagramme d'activité[1].

Un support de capitalisation et de partage de connaissances

« Support », *« capitalisation »* et *« partage »* sont des qualifications importantes de la carte géographique et de notre carte de connaissances biologiques. Ce paragraphe met en évidence à partir d'un cas d'utilisation hypothétique comment la carte revêt ces aspects et répond à des besoins pratiques. Le contexte est l'analyse de données d'expression de gènes concernant la leucémie humaine.

Après l'obtention de ses données d'expression, Pierre fait appel à un logiciel d'analyse de données et de regroupement automatique. Ce logiciel invoque alors l'extraction d'une carte auprès de l'entrepôt. Dans cet outil, le chercheur accède à la connaissance du domaine, masque les gènes qui lui semblent sans intérêt et met en valeur ceux qui lui semblent importants. Enfin, il ajoute quelques commentaires à certains éléments de données (gènes, …). Durant cette activité, il a, entre autres, consulté la littérature du domaine. Il a pris connaissance de certains articles liés à des gènes d'intérêt. Il a aussi pu mettre en relation certains gènes avec des articles de référence dont il avait déjà connaissance : ceci lui a permis de mettre en relief les gènes connus.

Plus tard, il souhaite revenir plus amplement sur sa bibliographie. Il exécute une application de recherche d'information, de veille scientifique et de gestion bibliographique. La carte lui permet alors directement de retrouver les articles dont il avait pris connaissance et qu'il avait retenus.

Un autre membre de l'équipe, Jean, charge la carte en parallèle dans l'application de données d'expression[2]. Il découvre de nouveaux documents pertinents et les met en évidence : Pierre en est informé via son application.

Pierre dans sa recherche d'information constate alors que certains documents ne correspondent pas à leur thématique. Il les ôte de la bibliographie. Une trace de l'utilisation permet de proposer deux alternatives à Jean : souhaite-t-il que les documents soient masqués automatiquement ou non ? Les gènes qui sont associés uniquement à ces documents doivent ils disparaître ou être grisés ?

Comme les pratiques de la biologie n'ont pas évolué vers le « tout numérique », de temps en temps, Pierre et Jean expriment le besoin de capturer un instantané de cette carte, et de le sauvegarder. L'image

[1] Cf. page 217 pour une présentation de ce formalisme.

[2] Le document est partagé et synchronisé au travers du réseau, la section suivante montre comment cela est implémenté au niveau logiciel.

capturée est par la suite imprimée et collée sur le cahier d'expériences. Pierre et Jean peuvent alors la griffonner ensemble autour d'un café ou l'emmener sur la paillasse.

Au travers de la carte, l'utilisateur est propriétaire des données ; ces données n'évoluent pas sans son consentement et il peut les mettre à jour s'il le souhaite. Cette carte peut être exportée vers des fichiers dans divers formats : SQL, XML, RDF, OWL, etc. Le format SQL permettra le chargement dans un SGBDR. Enfin, à partir des applications clientes, il est possible d'obtenir des captures d'écran (haute définition) qui sont par la suite imprimables. La carte peut se matérialiser de façon multiple ; elle est un « instantané » de la connaissance d'un domaine. L'équipe de recherche peut ainsi être propriétaire d'une carte et interagir avec, en capitalisant les connaissances collectives : ils annotent des gènes, commentent la bibliographie, filtrent les informations non pertinentes ou non fiables, etc.

Une fois propriétaire, il est possible d'importer ou mettre à jour les données, mais ceci se fait sous le contrôle de l'utilisateur, au sein des applications clientes. La donnée est alors au centre des différentes applications métier. L'utilisation du SGBDR amène directement la capacité de partage et de la synchronisation des applications clientes et des utilisateurs.

Diagramme d'activité détaillé

La figure 4.9 propose une représentation plus détaillée du fonctionnement de l'environnement. Les procédures relatives au niveau serveur et au niveau client sont décrites chronologiquement et parallèlement.

Au niveau serveur, l'entrepôt est construit par une succession de procédures. Les données sources sont intégrées successivement. Par la suite, toutes les données textuelles sont analysées (titres, résumés, synthèses, définitions, etc.). Un index associe de façon ordonnée chaque document aux concepts qu'il contient. Une analyse distributionnelle succède alors (fréquence, colocations, cooccurrence, etc.). On peut alors appliquer des méthodes d'extraction d'information sur les données de l'entrepôt, en utilisant, si besoin est, l'index et les mesures statistiques calculées antérieurement. Toute cette phase de construction de l'entrepôt est détaillée dans le chapitre 5.

Parallèlement au niveau du client, l'utilisateur est amené à lancer une application à sa disposition. L'utilisateur nécessite alors de récupérer les données. S'il dispose d'une carte existante (récupérée antérieurement ou par l'intermédiaire d'un collaborateur), il lui suffit d'en spécifier l'accès au sein de l'application cliente (chemin, IP, etc.). Dans le cas contraire, l'application cliente invoque l'extraction d'une carte auprès de l'entrepôt.

Nous ne proposons aucune définition restrictive pour la notion de contexte. Nous n'avons par conséquent pas formalisé la notion de service d'extraction et laissons ouverte la définition d'une méthode d'extraction de carte. L'environnement spécifie quelle méthode utiliser et expédie les paramètres qui en dépendent. Pour illustrer la diversité des possibilités, prenons deux exemples d'applications courantes. La première concerne une recherche bibliographique. Une requête est basée sur des mots clefs structurés par des opérateurs booléens. La réponse souhaitée par l'application cliente consiste en un corpus de documents avec les auteurs, mots clés et revues. Une seconde application d'analyse de données d'expression dispose initialement d'une liste de gènes et du nom de l'organisme. L'objectif est alors de récupérer les informations relatives aux gènes : annotations (concepts), documents, protéines. Il n'est pas forcément nécessaire de stocker dans la carte les données de séquences, ou encore les nœuds correspondant à des auteurs et revues pour diminuer le volume de la carte. Le chapitre 6 détaille nos propositions concernant l'adaptabilité et la contextualisation.

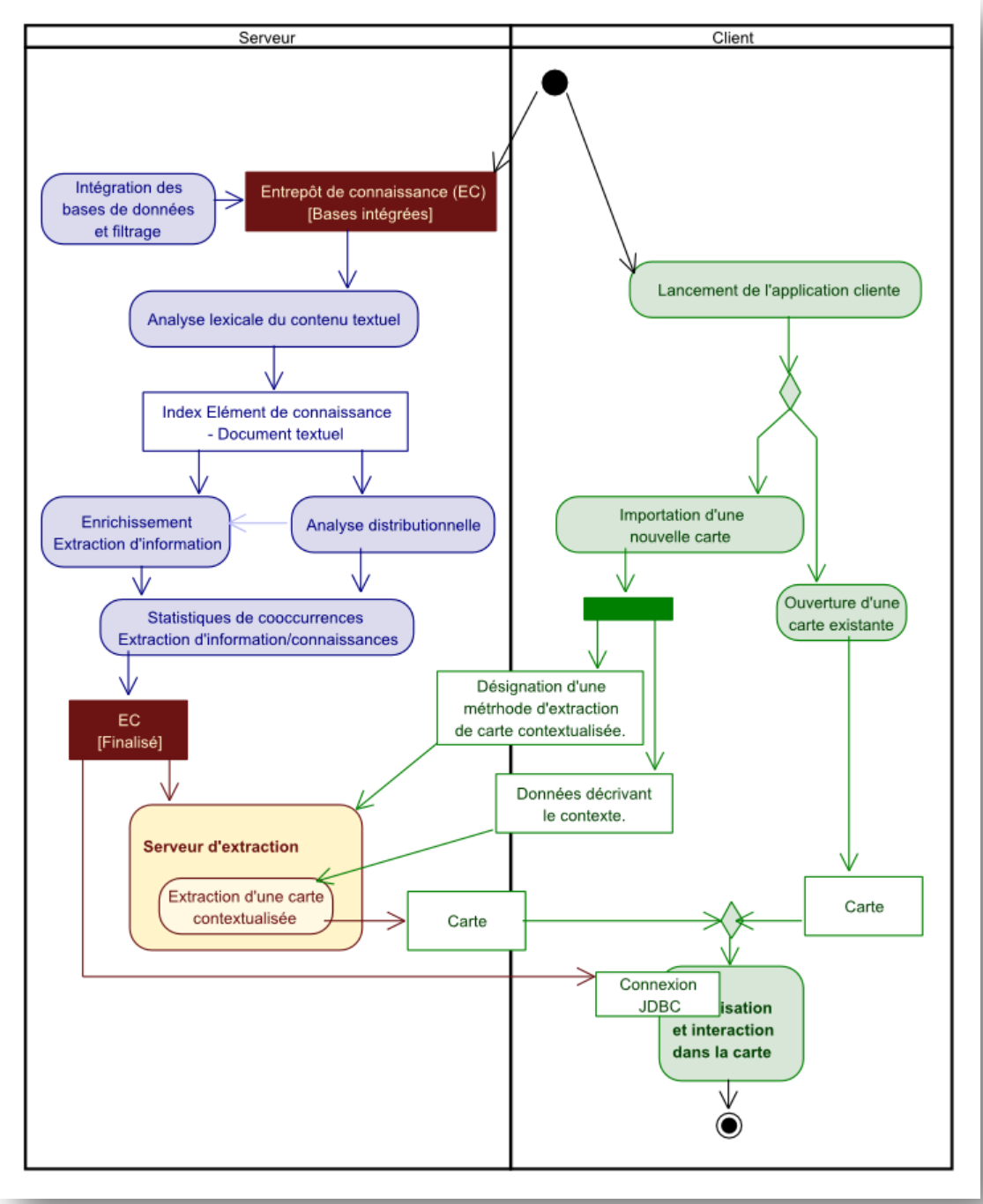


Figure 4.9 – Diagramme d'activité

4.3.3 Architecture logicielle

L'architecture logicielle est détaillée dans la figure 4.10. Nous distinguons deux niveaux : le niveau de persistance est celui du serveur de données, le SGBDR. Le second niveau est celui du client et de l'utilisateur. Lorsque l'utilisateur utilise une carte sous forme de fichier, il n'y a pas de serveur. L'accès au serveur se fait via le langage de requête SQL. Des procédures en PL/SQL permettent de gérer la traçabilité de façon transparente au niveau client et améliorent aussi les performances.

La manipulation des données par le client se fait via une API orientée objet dont nous avons présenté le modèle. Cette API utilise une interface (le « connecteur ») lui permettant d'accéder indifféremment au contenu du fichier ou du SGBDR. Il est possible de créer des connecteurs composites pour envisager l'accès simultané à plusieurs sources.

Les procédures de construction de l'entrepôt (intégration, analyse lexicale, enrichissement) sont présentes au niveau du serveur. Localement, ces procédures s'exécutent en effet nettement plus rapidement. Les procédures utilisent tantôt l'API pour un développement plus rapide, sûr et facile, tantôt un accès direct en SQL et PL/SQL dans le souci des performances.

Le développement du client, au contraire se fait par l'intermédiaire de l'API. Il est ainsi possible de systématiser l'usage des opérations métiers et la traçabilité qui en découle. Le développeur dispose d'une couche graphique permettant l'implémentation simple et rapide d'interfaces utilisateurs riches et adaptées à des tâches spécifiques. Cette couche est construite à partir de la boîte à outils Prefuse [Heer, Card et al. 2005].

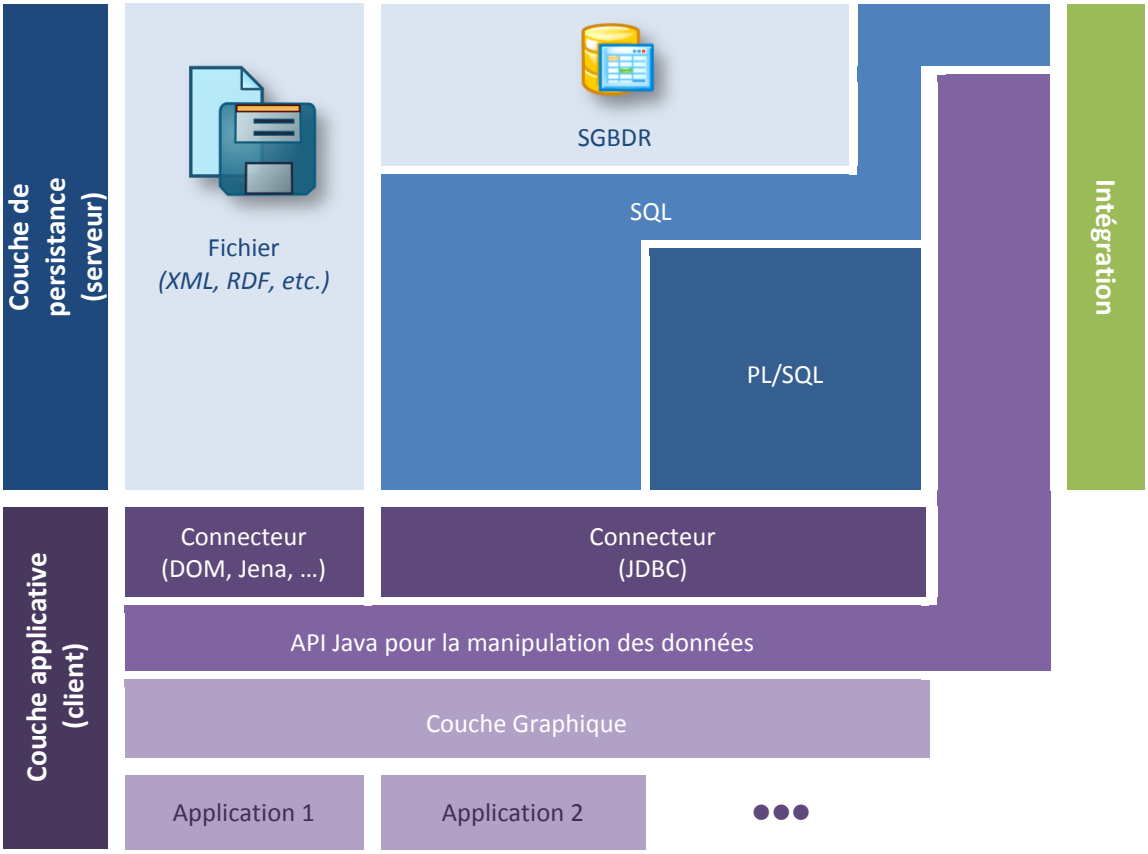


Figure 4.10 – Architecture logicielle

4.4 Synthèse

Nous venons de présenter l'environnement I²DEE suivant différents points de vue : l'architecture logicielle, le modèle, les caractéristiques, etc. Nous avons ainsi mis en évidence sa souplesse et son extensibilité du point de vue du développeur, que celui-ci construise une nouvelle application ou souhaite étendre une application existante.

Le fonctionnement du point de vue de l'utilisateur reprend le principe général de la carte : à partir de données, on génère une carte concernant une région donnée de l'information (organisme, domaine) et adaptée à une problématique particulière (analyse de donnée, recherche bibliographique, etc.).

On distingue globalement trois grandes composantes logicielles dans l'environnement qui s'articulent autour du modèle et de l'API que nous venons de décrire : les procédures d'intégration, la couche graphique permettant la construction rapide de clients riches, et les mécanismes d'extraction de carte contextualisée et d'adaptabilité. Ces composantes sont détaillées dans les trois chapitres qui suivent.

L'architecture comme le modèle sont simples et ouverts. On peut ainsi espérer qu'I^{2}DEE se développe au sein de la communauté. Cette approche se positionne comme unificatrice : elle permet réellement à la plupart des solutions existantes de s'interfacer avec, à moindre coût. L'intégration d'une nouvelle source dans I^{2}DEE serait ainsi directement exploitable au sein d'un système d'intégration, d'un service Web, consultable depuis un portail, et accessible au sein des applications métiers de l'utilisateur. I^{2}DEE est donc un environnement capable de mutualiser et structurer les efforts de la communauté et de permettre une réutilisation immédiate des produits de ces efforts.

CHAPITRE 5

Construction de l'entrepôt

« From this perspective, the significance of the "information explosion" may lie not in an explosion of quantity per se, but in an incalculably greater combinatorial explosion of unnoticed logical connections. Science responds to growth by increasing specialization, but tends to neglect connections. [...] This emergent property of very large literature bases – the information mosaic of science – presents unique problems and unique opportunities. »

DON R. SWANSON

Afin de décrire le déroulement procédural des différentes étapes d'intégration des données, nous avons choisi d'employer UML. Cette norme est le principal standard de modélisation orientée objet. Le formalisme qui correspond plus particulièrement à notre besoin est le diagramme d'activité. Nous l'utilisons à un niveau macroscopique pour décrire un système ou au niveau interne d'un objet pour décrire son comportement, un algorithme, etc. Répandu et structuré syntaxiquement, UML donne souvent lieu à des interprétations sémantiques variées. Une présentation détaillée du formalisme des diagrammes d'activité d'UML est proposée dans l'annexe D.2 (page 239). Les couleurs n'ont pas de signification formelle, elles facilitent l'appréhension des diagrammes et le repérage de certains objets récurrents (notamment l'entrepôt).

5.1 Introduction

Dans le chapitre précédent (page 130), nous avons décrit l'architecture générale de l'environnement I²DEE (figure 4.9 page 130) qui permet d'intégrer des données hétérogènes, de les partager et de les visualiser sur différents supports. Nous distinguons deux phases essentielles au niveau du serveur : la construction de l'entrepôt de connaissances **(EC)** et la mise en œuvre d'un service d'accès et d'extraction contextuelle de cartes. Ce chapitre détaille la première phase de laquelle dépendra la suite.

Il est difficile de proposer une méthode permettant plusieurs intégrations indépendantes et concurrentes. Une grande partie des ressources possède des dépendances vis-à-vis d'autres ressources. En tenant compte de ces dépendances, nous proposons une méthodologie d'intégration des ressources dans l'entrepôt. Elle se décompose globalement sous forme de trois étapes successives :

- intégrer les bases de données généralistes référencées par les portails du domaine,
- ajouter les bases de données du domaine,
- enrichir l'entrepôt par l'analyse des données textuelles, la mise en œuvre de méthodes d'extraction d'information ou la fouille de données.

Dans la suite, la présentation de chacune de ces étapes est illustrée à l'aide de diagrammes d'activité UML (cf. annexe B.2 page271). Les sections qui suivent détaillent les procédures d'intégration et d'enrichissement. Enfin, une section complète présente le fonctionnement de l'analyseur lexical que nous avons développé et détaille l'analyse distributionnelle réalisée. Nous discuterons alors les résultats obtenus en mettant en relief les limites de cette intégration et les perspectives qui s'ouvrent à nous.

5.2 Vision générale de l'intégration au sein d'I²DEE

La figure 5.1 ci-dessous détaille l'organisation de la constitution de l'entrepôt. Par rapport au diagramme précédent (figure 4.9 page 130), une première distinction apparait qui sépare deux ressources courantes (UMLS et PubMed) des bases de données du domaine. En effet, parmi les bases de données existantes, certaines sont fréquemment utilisées et représentent une référence pour une large partie de la communauté. D'autres correspondent à des domaines plus restreints. Il est souvent difficile de classer les ressources dans l'une ou l'autre de ces deux catégories. Cependant, UMLS et PubMed concernent la totalité du secteur biomédical et sont détachés de toute application. La plupart des portails du domaine référencent leur contenu. Ces deux ressources se révèlent essentielles dans la constitution de l'entrepôt, quel que soit le domaine de recherche de l'utilisateur. L'architecture présentée dans le chapitre précédent montre qu'il est possible de manipuler les données au travers de l'API provenant de plusieurs sources. Cette possibilité permet de proposer différents services de cartes, plus ou moins génériques, et donc de mutualiser l'effort d'intégration le plus coûteux.

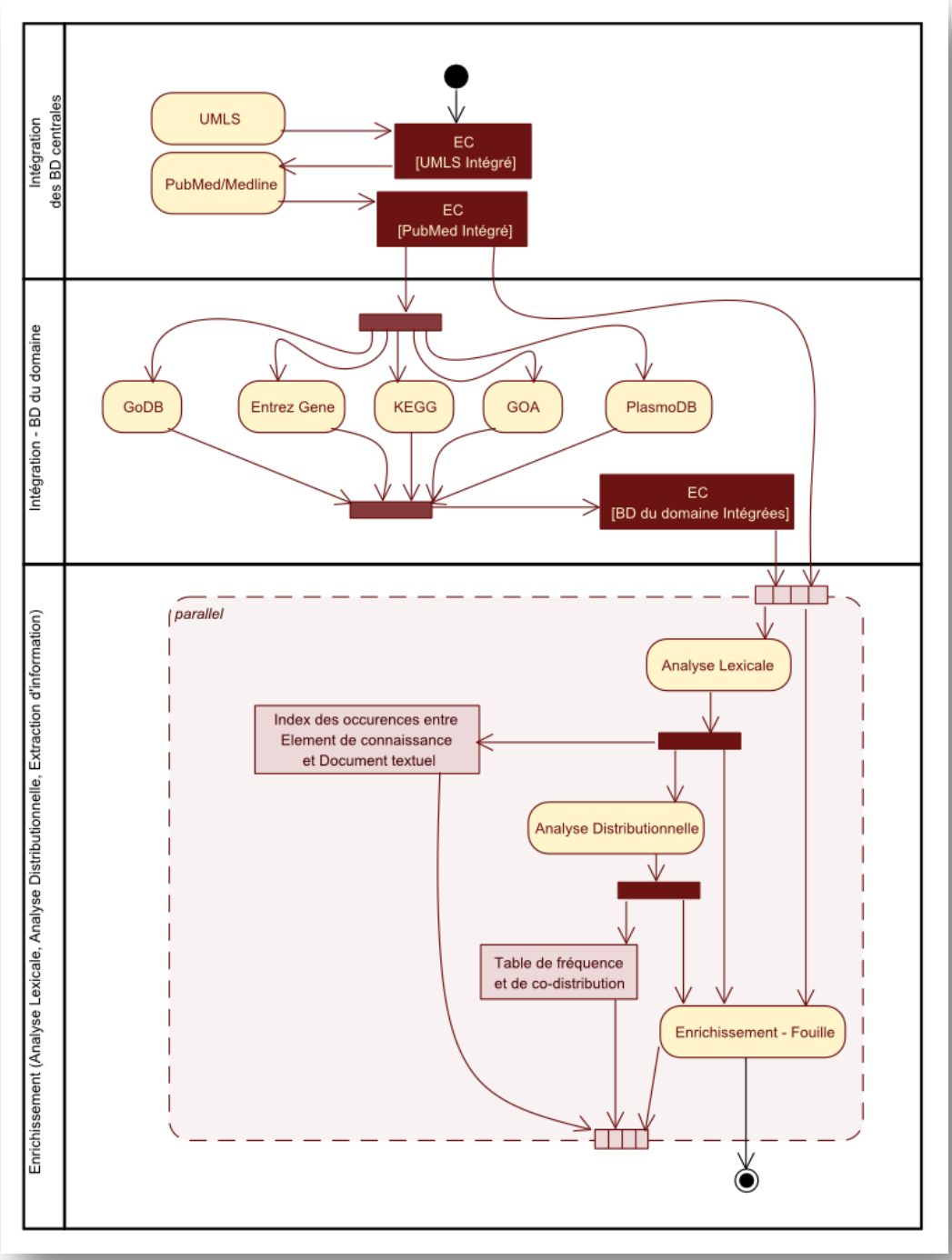


Figure 5.1 – Construction de l'entrepôt.

PubMed est une base de données qui centralise à l'échelle mondiale l'ensemble des publications de la communauté des sciences du vivant et du domaine médical. UMLS n'est pas directement impliqué dans le quotidien du biologiste, mais il contient, entre autres, deux ontologies de référence en biologie. La première, GO (Gene Ontology), est utilisée dans la plupart des bases de données pour annoter des gènes, protéines, etc. La seconde, MeSH (Medical Subject Headings), est un vocabulaire contrôlé support de l'indexation des documents dans PubMed. Elle est peu connue des biologistes qui l'utilisent en toute transparence au sein du portail de PubMed. Dès lors, UMLS, qui contient plus d'une centaine d'ontologies, s'impose indirectement comme une ressource essentielle à la gestion de l'information manipulée par les scientifiques.

Ces arguments justifient la présence de PubMed et UMLS dans le processus d'intégration. Leur position est justifiée par une dépendance entre les ressources. UMLS contient les ontologies, supports de l'information pour la plupart des systèmes d'information biologique existants. En contrepartie, il ne nécessite aucune ressource *a priori* pour s'intégrer dans l'entrepôt. Il prend donc naturellement place en première position dans l'entrepôt. PubMed dépend essentiellement d'une seule ressource, le MeSH. En contrepartie, de nombreuses bases de données référencent PubMed. Par exemple, PlasmoDB propose pour ses annotations des références vers les articles source de cette indexation. PubMed prend donc naturellement la suite directe d'UMLS dans le processus d'intégration. Suivent alors les bases de données du domaine comme KEGG, GOA, PlasmoDB (pour *Plasmodium Falciparum*), PantherDB (pour les puces AB Systems) et TAIR (pour l'*Arabidopsis Thaliana*), etc. Leur ordre importe peu dans notre cas. Bien évidemment, en fonction des besoins, certaines dépendances peuvent à nouveau se manifester.

L'analyse lexicale et l'enrichissement de l'entrepôt succèdent à cette intégration. Cet ordre n'est pas strict mais recommandé. PubMed et UMLS ne sont pas modifiés par la suite. L'analyse lexicale des articles (résumés et titres) ou des définitions ne dépend pas des voies métaboliques de KEGG. Cependant, certaines de ces bases de données contiennent une information supplémentaire ou mettent à jour l'information présente dans l'entrepôt. Gene ajoute des noms de gènes, qui peuvent être exploités par l'analyse syntaxique et la génération de l'index. OMIM contient des synthèses concernant des affections chez l'homme. GODB met à jour certaines définitions. Les méthodes d'enrichissement peuvent donc dépendre de certaines ressources en fonction du contexte. De façon générale, il est recommandé d'attendre la fin du processus d'intégration pour procéder à l'enrichissement de l'entrepôt. Dans la suite de ce chapitre, nous détaillons le déroulement de chacune des procédures citées.

Précisons enfin qu'à l'origine de cette intégration, nous prévoyions d'intégrer KEGG. Cependant, nous ne l'avons pas prise en considération en raison du peu d'intérêt de nos collaborateurs biologistes pour cette ressource. Cette base de données nous semble cependant importante dans certaines applications et si elle parait assez méconnue d'une partie de la communauté, elle devra sans doute être intégrée pour répondre à de nombreux besoins.

5.3 Procédures d'intégration des bases de données

5.3.1 UMLS

UMLS est un entrepôt rassemblant plus d'une centaine d'ontologies dans le domaine de la biologie et du médical. Son « Metathesaurus » constitue un médiateur permettant d'aligner ces ontologies entre elles. Il se décompose en 4 niveaux : les concepts, les termes, les écritures de ces termes (orthographes, formes fléchies, variations diverses, etc.) et leurs occurrences dans les ontologies. Dans cette intégration, nous avons fait le choix de ne conserver que les deux niveaux les plus extrêmes. Les concepts deviennent des nœuds dans l'entrepôt, les occurrences deviennent des alias. L'alias est donc une écriture possible d'un concept, liée éventuellement à une source, en conservant la notion de préférence.

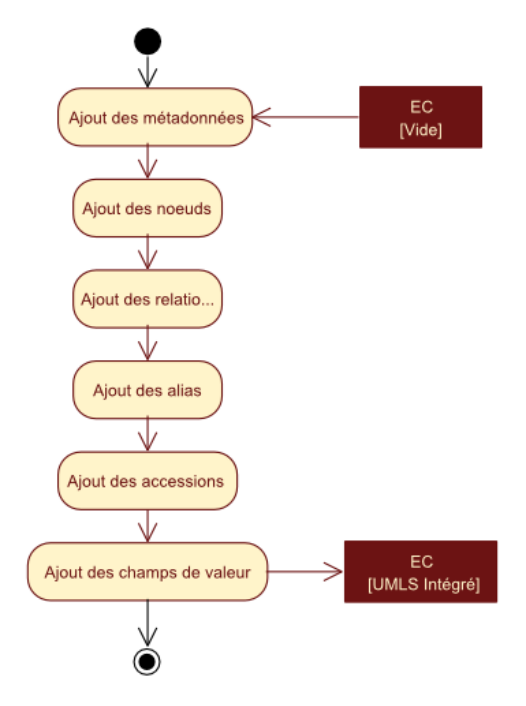


Figure 5.2 – Intégration d'UMLS

La procédure est organisée de façon linéaire (figure 5.2). On insère en premier les métadonnées (listes des sources, des types de relations sémantiques, des types de données, etc.). On ajoute ensuite les nœuds (concepts ou *types sémantiques*) avant les arêtes (relations).

UMLS propose en particulier deux types de nœuds. Le premier qui compose le Metathesaurus est le concept. Il y en a près d'un million ; ils sont reliés par plus de 60 types de relations différents. Ces dernières sont le plus souvent purement générales (essentiellement « est un » et « partie de »). D'autres, moins fréquentes sont parfois spécifiques à des domaines ou des ressources particulières (*« clinically similar »*, *« exhibits »*, *« dose form of »*, *« diagnosed by »*, *« manifestation of »*, etc.). Le réseau sémantique[1] d'UMLS repose sur des *types sémantiques* qui regroupent 135 concepts assez généraux hiérarchisés en 4 niveaux. Chaque concept d'UMLS est affecté à un *type sémantique*. On peut considérer cette démarche comme une classification ou encore comme un nouveau typage fin des nœuds. UMLS a choisi de présenter cela comme un mécanisme simplifié d'hyperonymie facilitant la mise en œuvre des mécanismes d'inférence et de fouille sur de grandes données. Dans cette optique, nous avons opté pour représenter les *types sémantiques* comme des nœuds distincts des concepts avec des relations d'hyperonymie (distinctes des précédentes) reliant un concept à son *type sémantique*.

Ensuite, nous insérons les orthographes de concepts dans les alias et les identifiants comme numéros d'accessions. Il existe près de 5 millions d'écritures différentes des termes. Notons qu'UMLS fournit directement la notion de préférence d'alias en lien avec la source. Enfin, nous ajoutons tous les champs de valeur correspondant aux données restantes (commentaires, définitions, etc.).

UMLS nécessite de nombreuses ressources. Le DVD que l'on installe en près de six heures nécessite près de 20 Go d'espace. Les données produites sont structurées dans un format plat adapté au chargement dans un SGBDR. L'intégration dure ainsi près de 30 heures, dont plus de la moitié est consacrée à la génération des index dans le SGBDR. Il est difficile d'envisager

[1] Semantic Network – *http://www.nlm.nih.gov/pubs/factsheets/umlssemn.html*

d'accélérer ou d'alléger fortement cette procédure, à moins de restreindre le nombre de sources. Les besoins cités précédemment considèrent l'installation de toutes les ontologies en anglais et avec les droits d'utilisation les plus ouverts (niveau 0). Dans le cas où l'on souhaite uniquement GO et MeSH, par exemple, l'installation se restreint à quelques heures. En outre, beaucoup d'ontologies sont plus spécifiques au domaine médical qu'à la biologie. Ce temps d'intégration n'est nécessaire que lors de la mise en place de l'application. Une fois intégrées, les données d'UMLS pourront être mises à jour régulièrement de façon incrémentale : UMLS propose 2 à 4 distributions par an avec une historisation systématique.

5.3.2 PubMed

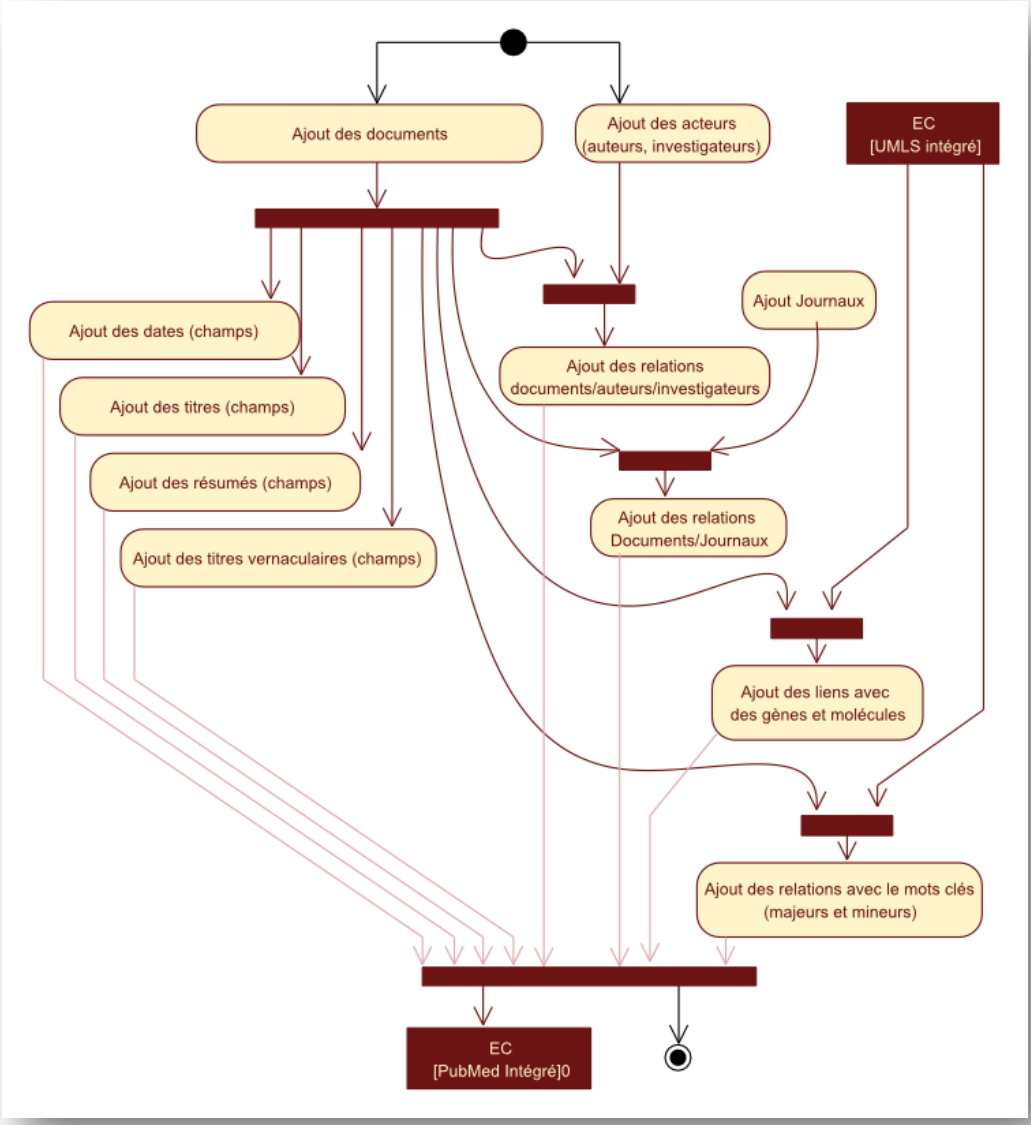


Figure 5.3 – Intégration de PubMed

Comme nous l'avons déjà présenté, PubMed, portail relatif notamment à la base Medline, est la ressource qui centralise à l'échelle mondiale les publications dans le domaine des sciences du vivant et du médical depuis les années 70. Il recense actuellement près de 14 millions d'articles provenant de 3000 revues différentes, produits par près de 44 millions d'auteurs distincts. La moitié des articles possède un résumé. Les relations correspondent aux relations entre un auteur et son document, entre un journal et ses articles, etc.

Les types de nœuds insérés sont donc les documents, les auteurs, et les journaux (figure 5.3). Les dates, titres, résumés, etc. se matérialisent sous forme de champs. Il reste alors deux données complémentaires : les entités biologiques liées aux documents (composés chimiques et gènes) et les mots clés (majeurs et mineurs). Les composés chimiques sont parfois déjà présents dans UMLS et sont reconnus. Lorsque le composé est absent ou qu'il s'agit d'un gène, un nœud est créé dans l'entrepôt avec l'alias correspondant. Les mots clés qui indexent un document proviennent de MeSH. Ils sont donc déjà présents dans l'entrepôt. La notion de « majeur » ou « mineur » est un attribut de la relation. Il est discutable de préférer la représentation de deux types de relations distincts et hiérarchisés. Nous relions ainsi les documents et leurs concepts indexés. Signalons tout de même que les gènes et produits chimiques indexés dans PubMed ont relativement peu d'intérêt dans notre cas. Nous prévoyons d'insérer les bases de données du domaine. Par conséquent, le lien entre entité biologique et publication scientifique est généralement contenu dans les bases du domaine avec plus de finesse, de précision et de fiabilité. Le volume occasionné par ces données est de l'ordre de 48 Go (décompressé), la durée d'intégration est assez longue (plusieurs jours en fonction de l'espace disponible en mémoire secondaire). Nous avons d'autre part utilisé BioText[1] qui convertit du format XML des archives de PubMed vers des fichiers plats plus pratiques à charger dans un serveur relationnel. Là encore, des mises à jour incrémentales de courte durée peuvent être mise en œuvre.

5.3.3 GODatabase

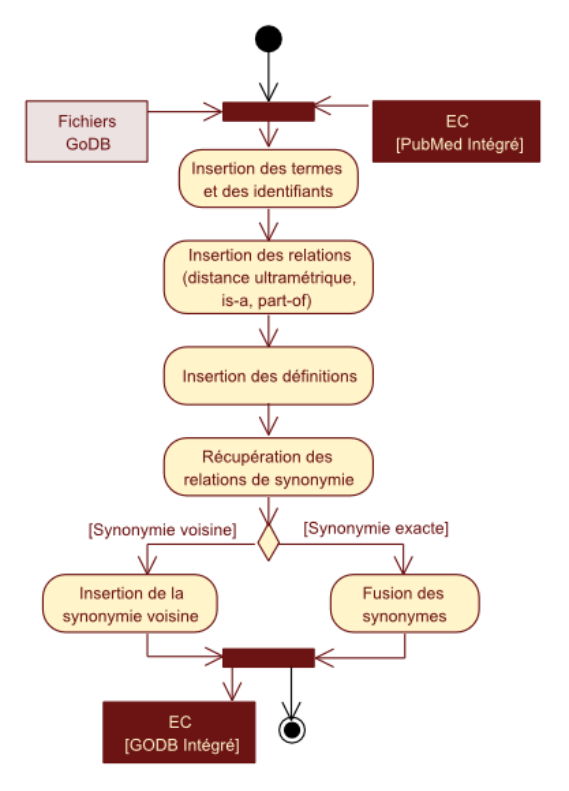


Figure 5.4 – Intégration de GODatabase

GODatabase (GODB) est la base de données produite par le *Gene Ontology Consortium*. GO est une ontologie particulièrement dynamique, construite notamment par la fusion d'autres

[1] *http://biotext.berkeley.edu/software.html*

ontologies. C'est comme cela par exemple que GO est passée en peu de temps (entre octobre 2006 et janvier 2007) de 14 000 à 20 000 concepts. UMLS n'est généralement pas à jour sur ces versions. En outre, GODatabase contient des données supplémentaires (figure 5.4), notamment une distance basée sur la longueur du plus court chemin séparant deux concepts. Cette distance pourrait être directement calculée, et n'a d'utilité que pour des applications spécifiques. Enfin, l'information la plus importante est certainement celle sur la synonymie. GO propose 5 nuances de synonymie (proche, lointaine, exacte, etc.). La synonymie exacte révèle des instances d'un même concept. GO fait le choix de les considérer comme des concepts distincts liés par cette relation. Nous choisissons de fusionner les deux concepts synonymes exacts. Nous n'avons pas mis en œuvre de hiérarchisation des autres nuances de synonymie, la nuance est une valeur (étiquette) de la relation.

GODB ne représente que quelques mégaoctets à télécharger. Son intégration ne prend que quelques minutes, les fichiers étant déjà dans un format plat tabulaire, prêt au chargement dans un SGBDR. Par ailleurs, d'autres fichiers sont disponibles concernant des annotations pour certains organismes. Ces annotations sont redondantes avec les bases de données GOA et PlasmoDB. Dans le contexte de notre étude de *Plasmodium Falciparum*, nous avons décidé de ne prendre en compte que les annotations de PlasmoDB, épurées (*« curated »*), qui devraient être les connaissances de la communauté les plus précises et les plus à jour.

5.3.4 Entrez Gene

Entrez Gene, anciennement Locus Link, est une base de données qui a notamment pour objectif de centraliser (et standardiser) les noms de certains gènes et faciliter le référencement croisé entre les bases de données. Les informations qu'elle contient servent essentiellement à compléter la base d'accessions et d'alias de noms de gènes. Parmi les références croisées, elle contient quelques informations sur les protéines codées, et des annotations GO. Nous ne prenons pas en compte ces dernières pour les mêmes raisons que précédemment : nous considérons que la base de données du domaine sélectionnée par les biologistes est la plus appropriée.

5.3.5 PlasmoDB

PlasmoDB est le portail génomique dédié à *Plasmodium Falciparum*. Il est basé sur le système d'intégration GUS. Il met ainsi à disposition de nombreuses informations génomiques. D'un point de vue plus fonctionnel, il propose aussi un petit résumé concernant le gène et leurs annotations. Ces informations sont extraites de GeneDB. Jusqu'ici, les fichiers étaient directement téléchargeables sur PlasmoDB sans une mise en évidence réelle de leur origine. Aujourd'hui, il n'y a qu'une référence vers une source de données. Alors que les séquences sont fournies dans des formats standards (Fasta), les informations fonctionnelles sont mises à disposition dans un fichier plat, similaire aux formats rencontrés précédemment.

5.4 Analyse lexicale

5.4.1 Motivations et choix

Le début de ce chapitre a présenté l'intégration des bases de données dans I²DEE. Parmi les ressources intégrées, nous retrouvons des bases de données bibliographiques, structurées en partie par le vocabulaire contrôlé MeSH, et les bases de données du domaine dont les informations fonctionnelles sont aussi structurées par une ontologie (GO). La structure sémantique générale du portail est donc assurée par la présence d'UMLS. Si on souhaite rechercher une information contenue dans un corpus textuel ou fouiller ce corpus, il est

nécessaire d'établir une correspondance entre son contenu textuel et les concepts de l'entrepôt de connaissance. C'est le rôle de notre analyse lexicale qui produit un index (ordonné) des occurrences de concepts dans les documents. Il existe autant d'index que de corpus (définitions, résumés, titre, synthèse OMIM, …).

On distingue différents types d'analyses reposant sur différentes techniques. La plus simple est le découpage[1], qui identifie les mots d'un document. Il est possible d'affiner ce découpage en identifiant pour chaque mot son lemme (lemmatisation) ou son radical (*stemmatisation,* francisation du terme *stemming*). La *stemmatisation* est basée sur un ensemble de règles permettant de supprimer des affixes (préfixes ou suffixes) [Porter 1980]. On dit que ce type d'analyse s'attache à la *morphologie dérivationnelle* du mot.

La lemmatisation se base sur un dictionnaire et prend en compte la morphologie flexionnelle du terme. Elle associe à un mot son lemme (l'entrée du dictionnaire pour simplifier) : par exemple, *« synthétisons → synthétiser »*, *« cellules → cellule »*, … La précision de la lemmatisation repose sur l'utilisation d'un dictionnaire qui établit la correspondance entre le lemme et ses formes fléchies (féminin, pluriel, conjugaison, etc.).

L'analyseur syntaxique s'intéresse à la *syntaxe*, c'est-à-dire à la fonction des éléments constitutifs du discours. Les analyseurs syntaxiques produisent un étiquetage (ou balisage) indiquant la fonction d'un mot dans un groupe (gouverneur, adjoint) ou d'un groupe dans une phrase (groupe nominal ou verbal, sujet, complément d'objet ou circonstanciel, etc.). Enfin, les analyseurs morphosyntaxiques, les plus perfectionnés, s'attachent simultanément à la morphologie flexionnelle et à la syntaxe.

L'anglais étant d'un point de vue *flexionnel* plus régulier que le français, l'utilisation de stemmatiseur est généralisée. Dans notre contexte de corpus scientifiques, les affixes sont importants et participent à la sémantique du terme : *« KIN », « kinase », « kinetic », « kinetoplast », kinetochore », « kinetor », « kinetogen », « kinetogenin », « kinins », « kininogenase »* représentent des concepts très différents, alors qu'une stemmatisation peut aboutir au même radical *« kin »*. Nous nous sommes donc orientés vers une lemmatisation, simple à mettre en œuvre, performante, et suffisante pour nos besoins actuels. Les analyseurs syntaxiques sont généralement moins fiables, perturbés par la présence de ponctuation au sein du terme, et bien plus lents. Ces constats sont issus d'expérimentations réalisées sur SYGMART, Synapse Cordial, Syntex et les SPECIALIST NLP Tools distribués avec UMLS et conçus par la NLM/NIH.

[1] Le terme original *« chunking »* est couramment employé à la place des termes français. La traduction littérale est le « morcellement ».

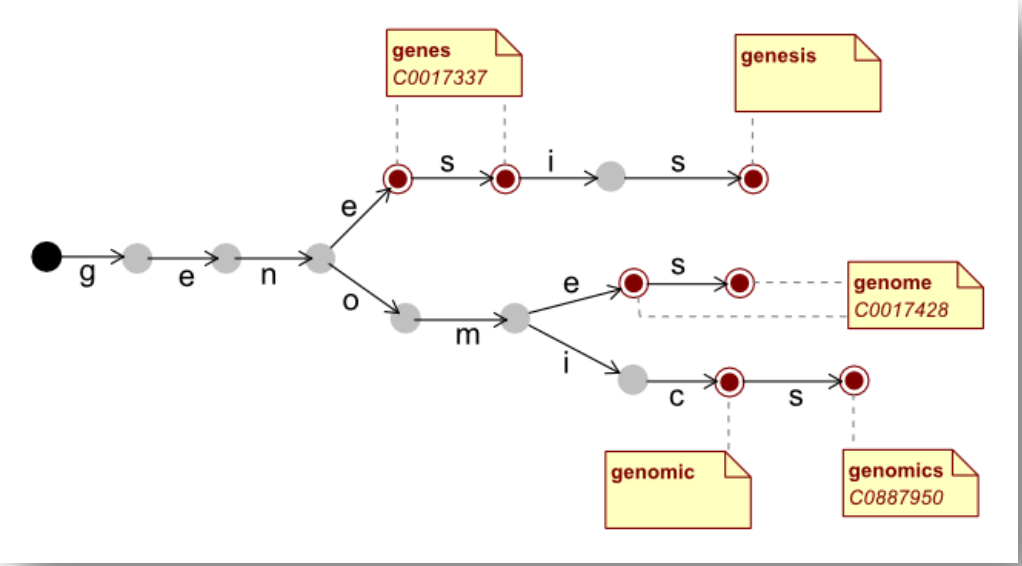


Figure 5.5 – Arbre à lettres – Cet automate possède une lettre pour chaque transition (flèche). Un état final (◉) marque la reconnaissance d'un mot ; il détermine un lemme (note). La racine correspond au mot vide. Ici l'arbre représente un dictionnaire contenant les mots : « gene », « genes », « genesis », « genome », « genomes », « genomic », *et* « genomics ».

5.4.2 Principe général : l'arbre à lettres

Dans un premier temps, nous devons lemmatiser le texte, c'est-à-dire reconnaître chaque forme fléchie d'un terme et lui associer le lemme correspondant. Nous utilisons pour cela une structure de données appelée « arbre à lettres » (figure 5.5) que nous avons déjà mise en œuvre par le passé [Jalabert, Munier et al. 2002; Jalabert 2003]. Cette structure proche de la notion d'automate à états finis est adaptée à la recherche d'une chaîne de caractères dans un texte [Crochemore, Hancart et al. 2001]. L'état initial est la racine de l'arbre. Un état final indique la reconnaissance d'une forme fléchie ; cet état référence un lemme. Chaque transition correspond à la reconnaissance d'une lettre supplémentaire dans le texte.

La lemmatisation découpe le texte en séquence de lemmes. Chaque élément de la séquence doit être le plus long possible. Statistiquement, on constate que ce choix est majoritairement pertinent. Cependant, il n'y a aucune certitude. L'exemple ci-dessous (figure 5.6) montre trois alternatives possibles pour une chaîne de caractères. Notre lemmatiseur privilégie la première solution (en gras). Pourtant, on peut imaginer des contextes où les autres alternatives seraient correctes. Nous justifions notre choix par l'hypothèse que la lecture est guidée par l'associativité de la mémoire et que le comportement du lecteur est comparable à celui de l'analyseur. Il parcourt le texte linéairement et reconnait de façon semi-globale des segments textuels. La reconnaissance d'une locution, issue d'un usage, est privilégiée par rapport à celle de plusieurs termes ; elle fera sens plus facilement. Ce choix devrait être remis en question dans le contexte de langues ayant un sens de lecture différent.

« … une pomme de terre cuite … »

[un] [pomme de terre] [cuit]	**(1)**
[un] [pomme] [de] [terre cuite]	(2)
[un] [pomme] [de] [terre] [cuit]	(3)

Figure 5.6 – Exemple d'ambiguïté liée à la recherche séquentielle du « lemme le plus long ». Ce choix semble intuitivement intéressant mais discutable en fonction du contexte.

L'ambiguïté précédente est purement sémantique : hors du contexte, il est difficile de privilégier assurément l'une de ces trois solutions, bien que la première soit plus fréquente.

Notre lemmatiseur introduit des erreurs qui pourraient être levées par une analyse syntaxique. Malgré des justifications linguistiques, certains cas mènent notre analyseur à l'échec, comme illustré dans l'exemple suivant (figure 5.7). Ici, c'est la disparition d'une partie de l'information syntaxique (tiret) et l'ignorance de la nature verbale de « porte » qui en sont la cause. Cet exemple montre les limites de ce type d'analyse ; cependant, dans la pratique, ces phénomènes sont peu fréquents, en particulier dans les corpus anglais spécialisés.

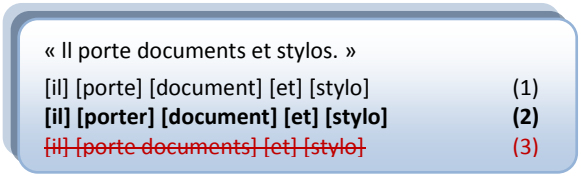


Figure 5.7 – Exemple de phrase dans laquelle la recherche du lemme le plus long mène à une erreur. L'ambiguïté peut ici être levée par une analyse syntaxique (il manque un syntagme verbal).

Le lemmatiseur est basé sur une structure d'arbre à lettres pour des questions de complexité algorithmique. L'algorithme recherchant les sous-chaînes les plus longues est de complexité linéaire en fonction de la taille du corpus. La limite de cette représentation est le volume de mémoire qu'elle nécessite pour le système, et qui dépend de la taille du dictionnaire. Dans notre cas, il y a près de 2 millions de termes de longueur moyenne élevée (près de 33 caractères). L'arbre à lettre « naïf » de l'automate va ainsi posséder plus d'une dizaine de millions d'objets. Ceci dépasse les capacités actuelles de la mémoire de notre station travail[1]. Nous avons ainsi été amenés à développer une méthode plus complexe permettant de pallier cette limite. Cette méthode normalise et filtre le contenu textuel (lexique et corpus) afin de réduire le nombre d'éléments du dictionnaire. De plus, elle recode tous les mots en les identifiant par un entier.

[1] Java, Windows XP, 32bits

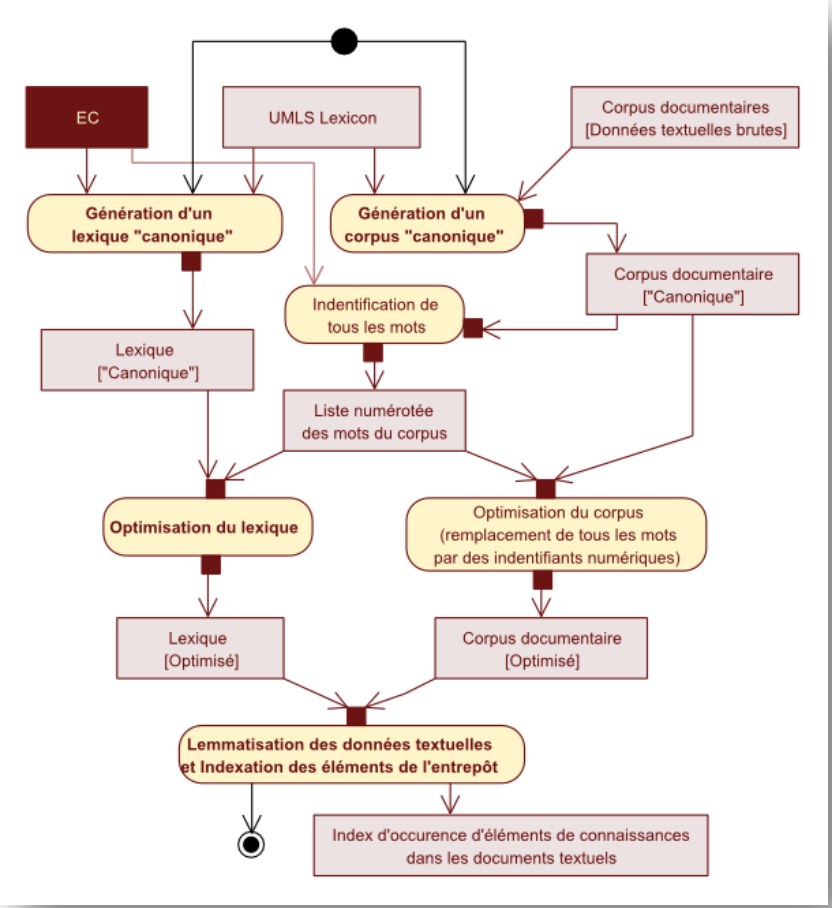


Figure 5.8 – Présentation générale de la lemmatisation. Les actions signalées en gras sont développées dans des sous-diagrammes par la suite.

Le diagramme ci-dessous présente le processus d'une façon générale (figure 5.8). Chaque action en gras est détaillée dans un sous-diagramme. Trois ressources sont utilisées initialement : la liste des termes et entités nommées de notre entrepôt de connaissances, le Specialist Lexicon (un dictionnaire de lemmes libre, pour l'anglais diffusé avec UMLS et les NLPTools), et le corpus documentaire à analyser. On distingue globalement plusieurs grandes étapes illustrées dans la figure 5.9) :

- la mise en forme canonique[1] du lexique et des corpus, qui permet de supprimer certaines variations (casse, ponctuation, blancs, …),
- la suppression des variations flexionnelles en lemmatisant les mots (et non les termes) dans le lexique et le corpus,
- l'optimisation du lexique et des corpus en recodant les mots par des identifiants numériques,
- la génération de l'index associant à un document une liste ordonnée de concepts de l'entrepôt.

[1] Cette appellation n'est pas un terme usuel du domaine, elle rejoint la définition générale proposée par Wikipédia : *« La mise en forme canonique est le procédé par lequel on convertit des données qui ont plusieurs représentations possibles vers un format 'standard'. On en éprouve généralement le besoin lorsque l'on veut pouvoir faire des comparaisons logiques, pour améliorer l'efficacité de certains algorithmes en éliminant les évaluations superflues, ou pour permettre d'ordonner des éléments en fonction de leur sens. »*

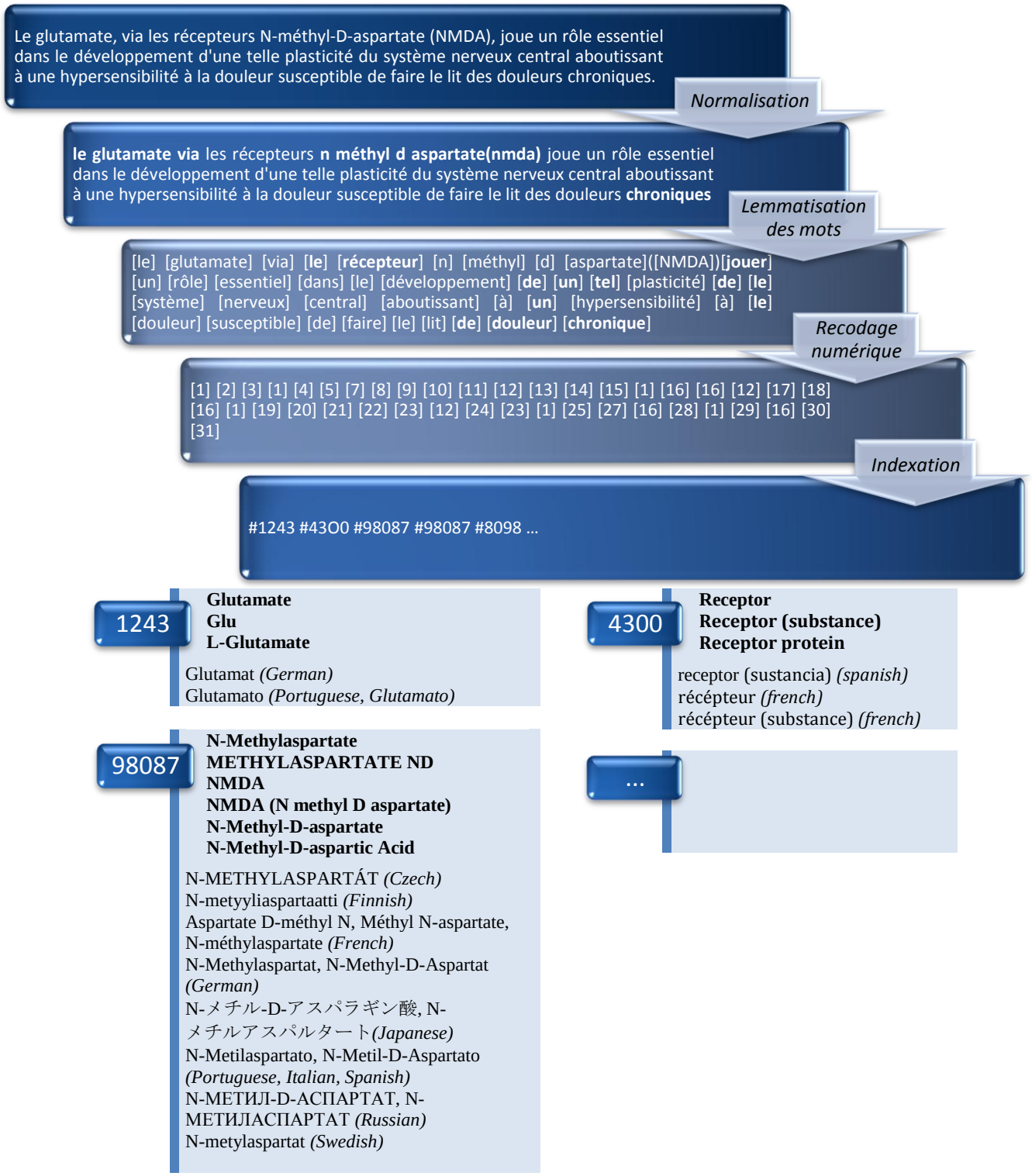


Figure 5.9 – La première partie de cette figure montre les transformations successives permettant d'aboutir à une séquence de concepts à partir d'un texte. La seconde illustre la correspondance entre un concept identifié par un numéro dans notre entrepôt et dans le texte. Les différents termes des concepts sont détaillés.

5.4.3 Mise en forme canonique du lexique et du corpus

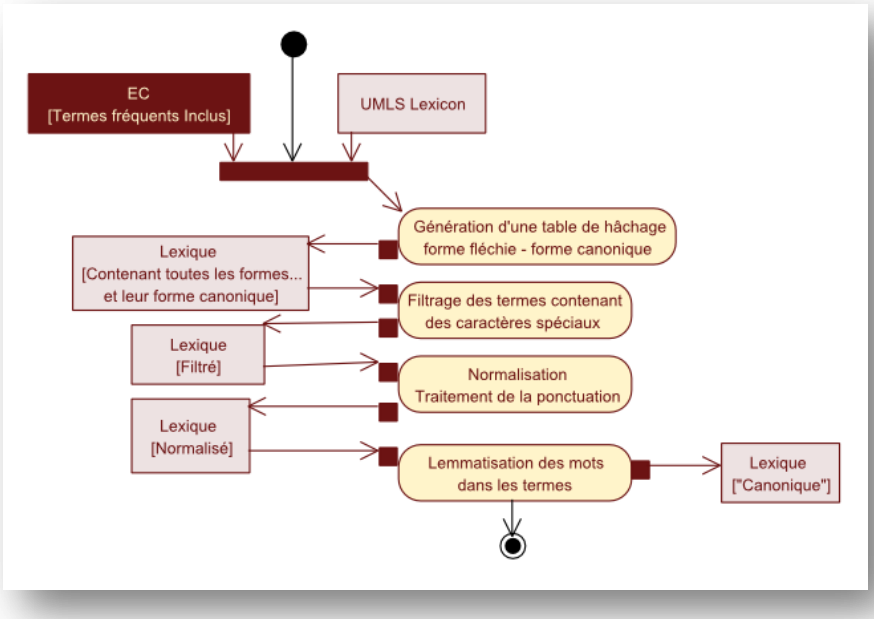


Figure 5.10 – Analyse lexicale – Normalisation du lexique

La constitution du lexique normalisé se découpe, elle aussi, en plusieurs étapes (figure 5.10). On commence par insérer dans une table de hachage (ou un index) les correspondances entre les formes fléchies et les lemmes contenues dans le Specialist Lexicon. On ajoute ensuite les éléments de l'entrepôt (termes, entités nommées, etc.). On filtre alors les termes contenant des caractères spéciaux invalides. La normalisation permet de gérer des variations graphiques des termes comme le montre la figure 5.11.

Figure 5.11 – Exemple de variation de l'écriture d'un terme chimique.

La normalisation consiste à :

- convertir les caractères en minuscules
- supprimer les caractères interdits
- homogénéiser tous les espaces blancs, réduire les espaces multiples, supprimer les espaces autour des virgules et parenthèses, etc.
- remplacer les tirets et les parenthèses par des espaces blancs
- lemmatiser les mots des termes
- supprimer les doublons qui seraient apparus à l'issue de cette procédure

On applique la lemmatisation des mots du terme. Ceci est sensiblement différent de la lemmatisation du terme, car on génère potentiellement une ambiguïté en faisant disparaitre une information permettant de lever l'ambiguïté. L'exemple de figure 5.12 illustre comment un segment textuel peut mener à plusieurs solutions. La normalisation peut aussi être source d'ambiguïtés (figure 5.13).

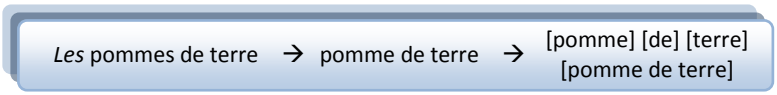


Figure 5.12 – Exemple d'ambiguïté due à la lemmatisation.

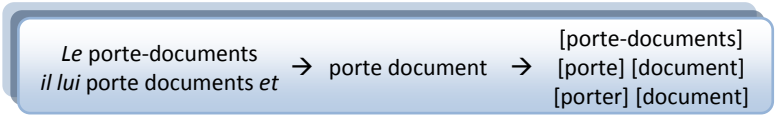


Figure 5.13 – Exemple d'ambiguïté due à la normalisation et la lemmatisation

Ces ambiguïtés ne sont pas toujours causées par la normalisation ou la lemmatisation. L'influence néfaste de la normalisation et de la lemmatisation sur la précision[1] est supposée négligeable. Ses bénéfices ne portent pas seulement sur les performances mais laissent espérer une amélioration du rappel[2]: si par exemple pour certaines molécules l'écriture du tiret n'est pas respectée, en généralisant cette hypothèse dans la normalisation, on peut espérer reconnaître plusieurs formes écrites d'une même molécule, qui ne sont pas présentes en l'état dans le lexique. Ainsi l'écriture des tirets dans les formules chimiques donne lieu à quelques variations, contrairement aux nombres et aux virgules par exemple qui sont rigoureusement respectés. Concernant le parenthésage, il n'est parfois pas nécessaire (cf. figure 5.14), mais il correspond à une formalisation rigoureuse. La question de supprimer ce parenthésage nécessite d'évaluer les gains de rappel et les pertes de précision occasionnés. Sachant qu'UMLS gère des orthographes différentes comme dans l'exemple ci-dessous, nous avons décidé de ne pas apporter de modification au parenthésage (si ce n'est la normalisation des espaces blancs).

1-((2-hydroxyethoxy)methyl)-6-(phenylthio)thymine
1-((2-hydroxyethoxy)methyl)-6-phenylthiothymine

Figure 5.14 – Exemples de variations observées dans UMLS sur l'écriture de formule et composés chimiques

La figure 5.15 ci-dessous détaille le processus similaire de mise en forme canonique du corpus (normalisation puis lemmatisation des mots).

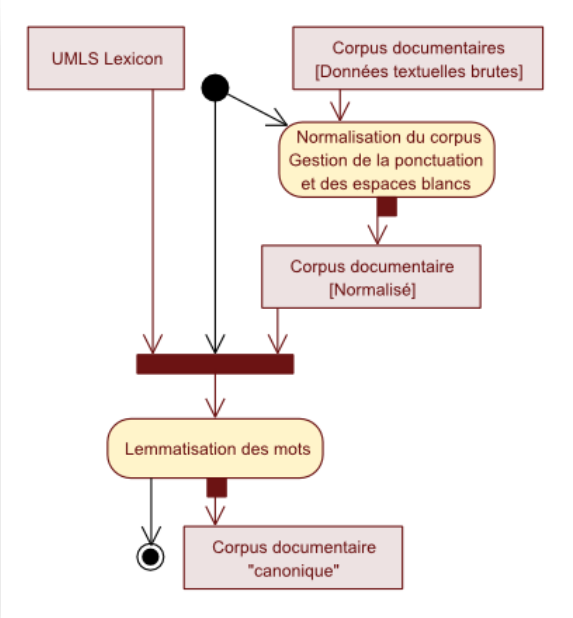


Figure 5.15 – Analyse lexicale – Normalisation du corpus

[1] La précision est le rapport entre le nombre d'éléments valides de l'ensemble réponse sur le nombre d'éléments de l'ensemble réponse. Une précision de 70% signifie donc que 30% des éléments de la réponse ne sont pas valides (ou pertinents) pour la requête donnée.

[2] Le rappel est le rapport entre le nombre d'éléments valides dans l'ensemble réponse et le nombre total d'éléments valides dans le corpus. Un rappel de 100% signifie donc qu'une méthode retourne la totalité des éléments corrects pour une méthode donnée.

5.4.4 Optimisation du lexique et du corpus avant la lemmatisation

Nous venons de présenter la mise en forme canonique du corpus et du lexique. A l'issue de cette procédure, certains termes du lexique du lemmatiseur sont devenus identiques. Nous supprimons les doublons. De plus, une procédure liste les mots qui ne sont pas utilisés dans le corpus. Tous les termes contenant ces mots sont supprimés dans le lexique du lemmatiseur. Il subsiste malgré tout un grand nombre d'objets en mémoire. Un second processus d'optimisation est mis en œuvre avant la lemmatisation et l'indexation à proprement parler. L'optimisation consiste à remplacer un mot par un identifiant numérique (figure 5.16). Un terme est alors représenté comme une succession d'identifiants numériques des mots qui le composent. Pour les molécules, acides aminés, gènes, etc. il faut ajouter les chiffres et le parenthésage qui font partie du terme. La même procédure d'optimisation est appliquée en parallèle sur le corpus. Les gains se situent à deux niveaux. Tout d'abord, un seul nœud est présent pour un mot et non une lettre, ce qui réduit de façon importante la taille de l'arbre. De plus, l'utilisation d'identifiants numériques permet de remplacer des objets par des types primitifs, optimisation liée au langage d'implémentation, Java.

5.4.5 Production d'un index

Une fois la séquence de lemmes obtenue, à chaque lemme correspond un élément de l'entrepôt de connaissances. Dans le cas où certains éléments sont absents de l'entrepôt, on les liste par ordre de fréquence. Ceux qui sont trop rares sont supprimés. Les plus fréquents sont filtrés manuellement (ou au travers d'une ressources bénéficiant d'une expertise)[1]. Les éléments restants sont ajoutés dans l'entrepôt (figure 5.17).

[1] Cette étape donne lieu à la création d'une *« stoplist »,* une liste de mots du langage qui ne sont pas pertinents (verbes modaux, prépositions, conjonctions, articles, etc.). Cette « *stoplist* » est créée manuellement, mais s'avère réutilisable dans le contexte d'une mise en œuvre élargie de l'environnement. Des *stoplist* existent par ailleurs déjà sur Internet.

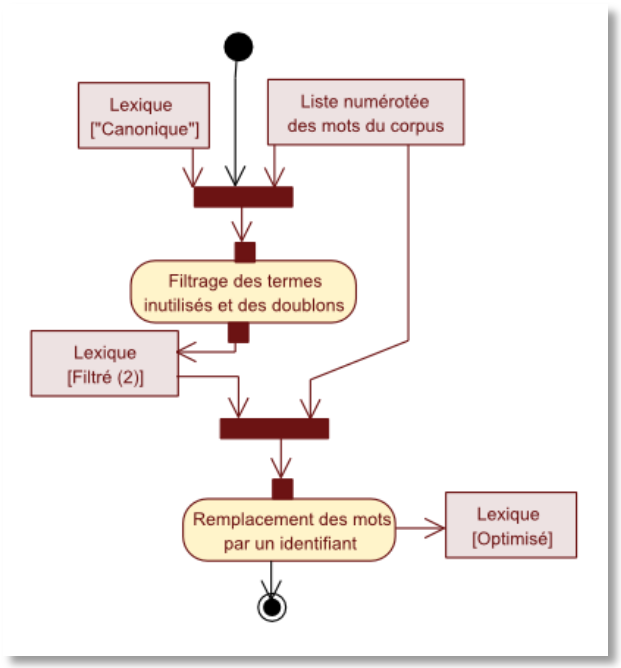


Figure 5.16 – Optimisation liée à l'implémentation de la structure.

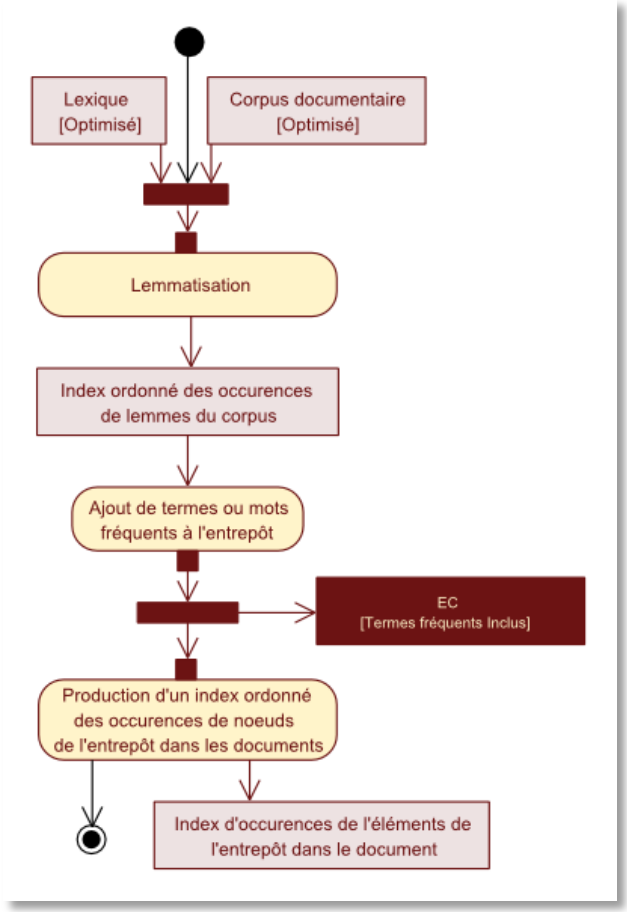


Figure 5.17 – Lemmatisation et production d'un index

5.4.6 Résultats et discussion

Nous n'avons pas réalisé d'évaluation quantitative. En effet, tout dépend du corpus : un corpus contenant de nombreux noms de molécules et d'acides aminés avantagera notre lemmatiseur. D'un point de vue qualitatif, nous arrivons à retrouver les noms de molécules, ce qui échouait dans les autres analyseurs. Nous n'avons pas trouvé d'ambiguïté introduite par le lemmatiseur. Du point de vue des performances, notre analyseur est bien plus rapide. Ceci est naturel puisque nous ne réalisons pas d'analyse syntaxique.

Concernant les performances, notre outil est bien plus rapide que les produits cités précédemment. Alors que les plus rapides nécessitent une minute pour analyser une centaine de résumés, notre lemmatiseur réalise cela en moins d'une seconde. Ceci est important : initialement pour le corpus PubMed complet (14 millions de titres, 7 millions d'abstracts), NLPTools était estimé à 160 jours de calculs, il nous faut maintenant moins d'une journée.

Après optimisation, il nécessite toujours une quantité de mémoire importante, mais réduite et raisonnable, de l'ordre de 50% des capacités de notre environnement (Windows, 32bits, Java). La construction du lemmatiseur nécessite quelques heures.

Plusieurs perspectives s'ouvrent à nous dans cette tâche d'analyse lexicale :

- utiliser un SGBD embarqué comme Apache Derby ou Oracle Berkeley DB nous permettrait d'éviter les phases d'optimisation, en dégradant les performances de l'analyse. Ceci peut s'avérer pertinent dans le contexte de petits corpus.

- combiner l'usage d'expressions régulières pour les formules chimiques, afin de limiter la taille du dictionnaire et de reconnaître des noms de molécules satisfiant la grammaire, mais absent du lexique. Le lexique ne peut être parfaitement exhaustif
- coupler cette analyse avec un analyseur syntaxique comme TreeTagger par exemple. En lui procurant un meilleur découpage, on peut espérer gagner en précision. Notre choix s'orienterait vers cet outil car il ne nécessite pas de dictionnaire, il repose sur une approche probabiliste, et il est traduit dans 13 langues.

5.5 Analyse distributionnelle

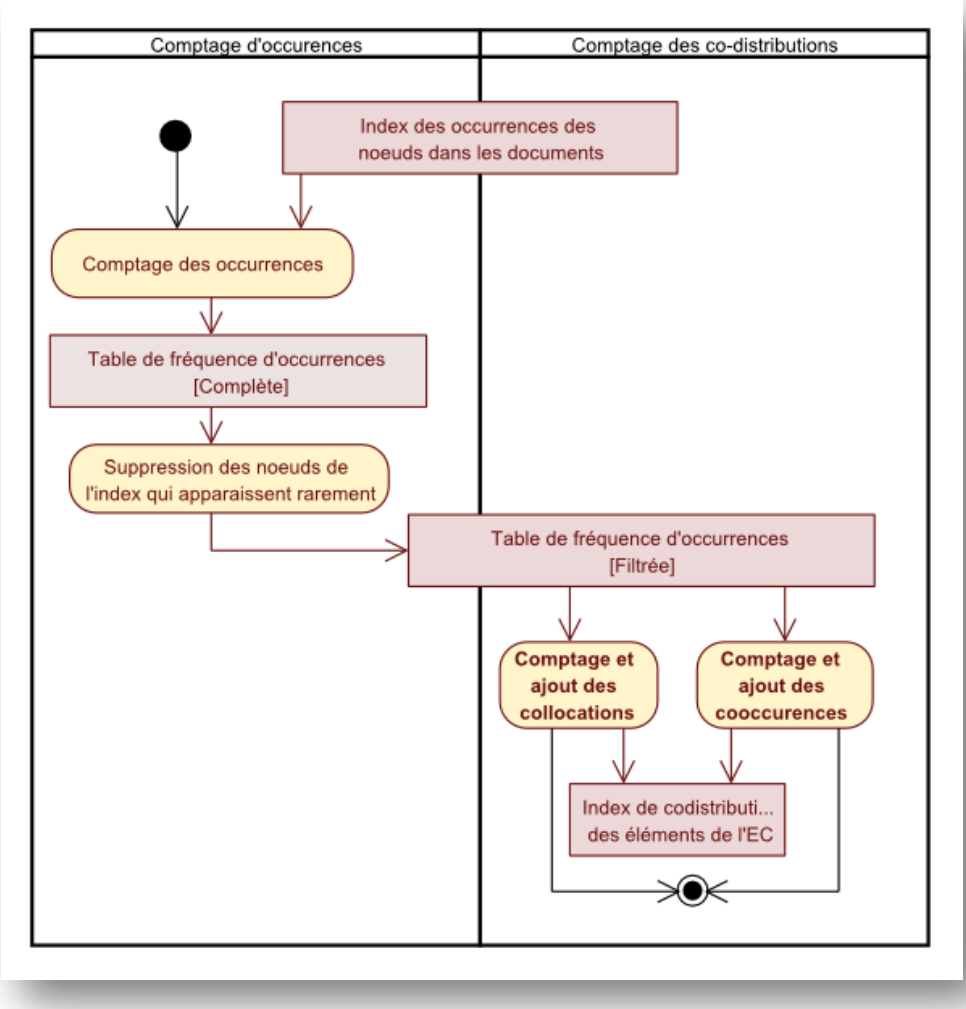


Figure 5.18 – Calcul des statiques distributionnelles : comptage des occurrences (fréquence) et de la codistribution.

A partir de l'index généré par l'analyse lexicale, nous établissons des statistiques de fréquence (figure 5.18). De ces dernières, nous sélectionnons les termes qui apparaissent au moins 3 fois[1] dans le corpus, et comptons les cooccurrences et collocations (deux notions que nous généralisons par le terme *codistributions*). Le calcul de ce type de statistiques est

[1] Cette valeur est arbitraire : en observant le corpus, un grand nombre de chaînes de caractères sans intérêt apparaissent une à deux fois. A partir de 3 occurrences, elles sont moins nombreuses. En contrepartie, certains termes pertinents mais très rares n'apparaissent pas plus de 3 fois. Cette valeur peut être modifiée et doit être choisie en fonction d'un compromis entre rappel et bruit.

généralement très lourd, en raison des besoins importants en mémoire. La complexité du calcul de cooccurrence est quadratique en fonction du nombre de termes. La collocation est bornée par le maximum entre le carré des termes et la taille du corpus. Dans la pratique, les collocations présentent peu de problèmes liés aux ressources nécessaires. Bien qu'elles nécessitent une grande quantité de mémoire, il est possible de les compter en une seule *passe*[1] (ou un nombre réduit de *passes*), en contenant la totalité (ou une forte proportion) des couples de termes en mémoire.

Le fonctionnement global de la procédure est décrit ci-dessous (figure 5.19). Pour chaque document, on parcourt les paires de termes consécutifs, on incrémente alors la valeur présente dans la mémoire tampon. A la fin de la procédure, on insère les données du tampon dans la base de données. Si la mémoire est pleine en cours de balayage des documents, on la vide complètement avant de reprendre (figure 5.20).

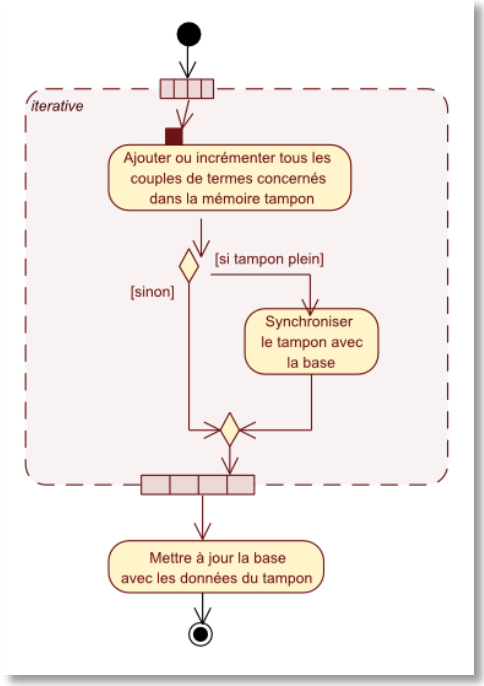


Figure 5.19 – Fonctionnement élémentaire d'un comptage, utilisé en particulier avec les collocations

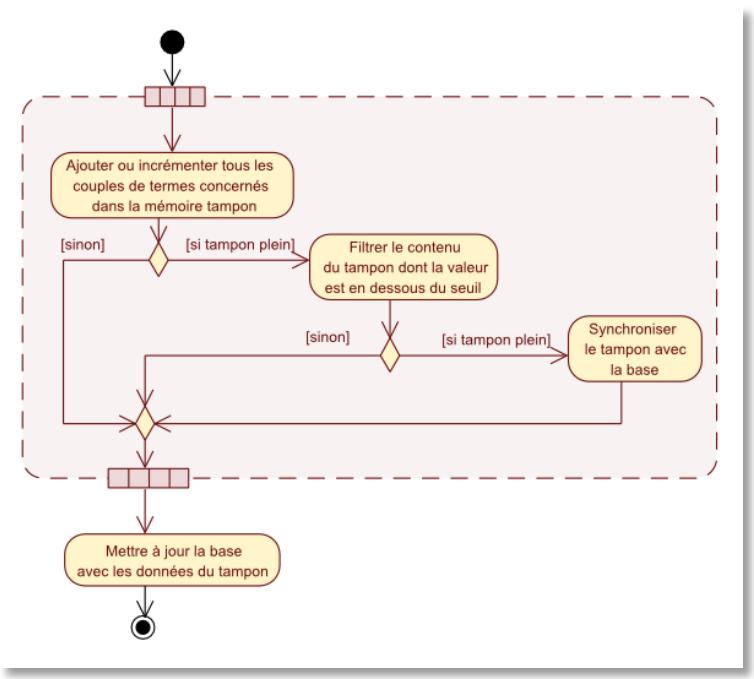


Figure 5.20 – Amélioration de la procédure par un filtrage de la mémoire permettant de limiter le nombre de vidage ou synchronisation entre cette mémoire et le SGBDR.

Concernant les cooccurrences, l'explosion combinatoire multiplie le nombre de *passes* et la durée de la procédure. La méthode précédente n'est plus applicable ; la mémoire sature rapidement. Lorsque la mémoire tampon est pleine, on supprime toutes les cooccurrences dont la valeur est égale à 1. Si le gain de mémoire est suffisant, on reprend directement, sinon on vide la totalité du tampon dans la base de données. On diminue grâce à cela le nombre de *passes*. En contrepartie, on introduit des approximations. Dans le pire des cas, lorsqu'on réalise n passes et qu'on remplit m fois le tampon, on réduit la précision de $n \times m$. Dans notre cas, n et m valent 3 et la perte de précision est donc limitée à 9 occurrences. Il est possible d'abaisser cette valeur en contrepartie d'une très forte hausse de la durée de la procédure ou dans le contexte d'autres

[1] La procédure consiste à lire des documents et remplir un mémoire tampon. Quand cette mémoire tampon est pleine, on la vide totalement dans la base de données avant de reprendre le parcours des documents et le remplissage du cache. C'est ce que nous appelons une *« passe »*. L'étape de synchronisation avec le SGBDR est particulièrement chronophage par rapport à la précédente. Dans le cas ou il n'y a qu'une seule synchronisation, on réalise directement des insertions sans vérifier l'existence d'une valeur précédente. Ceci est donc près de deux fois plus rapide.

architectures (systèmes 64bits, introduction d'une persistance intermédiaire discutée dans la conclusion de ce chapitre).

5.6 Synthèse

Nous venons de présenter l'ensemble du processus d'intégration lié à notre contexte d'application, une étude transcriptomique de *Plasmodium Falciparum*. L'essentiel des connaissances du domaine est centralisé au sein de PlasmoDB. UMLS et PubMed sont des bases de données généralistes nécessaires à toute (ou presque) étude concernant les sciences du vivant et le médical. Entrez Gene contient des données permettant le référencement croisé de bases de données. Enfin, GODB permet de compléter et mettre à jour les informations liées à Gene Ontology, au cœur du quotidien du biologiste.

Du point de vue des performances, les deux premières bases sont celles qui demandent le plus de ressources, tant en volume dans l'entrepôt qu'en temps de calcul durant leur installation et leur intégration. Elles nécessitent, sur une station de travail courante, de l'ordre de 3 à 4 jours et plusieurs dizaines de giga-octets pour le stockage. Les bases du domaine sont au contraire bien moins exigeantes et requièrent quelques dizaines de mégaoctets et quelques minutes tout au plus. Il faut cependant noter que *Plasmodium Falciparum* est un organisme n'ayant pas motivé de nombreuses contributions, possédant un petit génome. Dans d'autres contextes (humain, analyse interespèces, etc.), les problèmes peuvent être fortement accrus. Il faut noter que ce temps d'intégration n'est nécessaire que pour la mise en œuvre initiale de l'intégration ; par la suite, des mises à jour incrémentales sont possibles. Cette estimation est réalisée sur une station de travail courante. Enfin, cette procédure est *mutualisable* à l'échelle d'un laboratoire ou d'une communauté.

Il n'y a pas de limite particulière à l'enrichissement de l'entrepôt. Pour illustrer cette partie de la construction de l'entrepôt, nous avons mis en œuvre un analyseur lexical permettant d'indexer les termes contenus dans les données textuelles. Une analyse distributionnelle (fréquence, cooccurrence, collocation) est par la suite invoquée. Bien qu'aucune évaluation quantitative n'ait été menée, l'analyse lexicale présente (qualitativement) des résultats encourageants : elle pallie les limites que des autres outils qui sont inadaptés à de tels corpus, principalement parce qu'ils ne prennent pas en compte la ponctuation et les chiffres dans la syntaxe. Du point de vue des performances pures, malgré la dimension des données initiales, les méthodes implémentées permettent d'exécuter dans un temps raisonnable l'ensemble des procédures mises en œuvre. I²DEE montre ainsi la faisabilité, à l'échelle, de la construction de l'entrepôt. Un équipement courant et facilement accessible suffit. Une petite unité de recherche peut se constituer son propre entrepôt si elle le souhaite. Cela lui en coûtera, pour la mise en œuvre, quelques jours de calcul.

Gestion de la mise à jour et traçabilité

Nous avons montré la faisabilité de l'intégration, par des méthodes procédurales. Notre prototype ne met pas encore en œuvre de mise à jour des données. De plus, on ne conserve pas une partie des informations relatives à la traçabilité. Ceci est pourtant important pour un serveur en production. Il n'y a pas de limite particulière à ces démarches, si ce n'est le temps nécessaire à les implémenter, et la gestion des ressources qui comme nous l'avons évoqué déménagent régulièrement, changent de nom ou de format, etc.

Amélioration de l'analyse lexicale

Nos principales perspectives d'améliorations concernent l'analyse lexicale. La première, développée dans ce paragraphe, concerne le repérage de formules chimiques par des règles génériques. La seconde est liée à l'utilisation d'un niveau de cache supplémentaire et est décrite dans les paragraphes suivants. Nous souhaitons donc adjoindre un automate (expressions régulières) permettant de repérer les formules chimiques. La construction de celui-ci nécessite

l'apprentissage d'un dictionnaire de constituants des formules (« methyl », « hydroxy », etc.). Il serait préférable qu'un expert valide ce dictionnaire. UMLS puis PubMed semblent des outils précieux pour cela. Dans un premier temps, nous prévoyons que cette adjonction puisse améliorer le rappel de l'extraction de termes et de leur indexation : par exemple en reconnaissant des formules présentes dans PubMed, absentes d'UMLS et syntaxiquement valides. Cela permet d'enrichir l'entrepôt d'éléments jusqu'ici inconnus et d'améliorer l'indexation sous-jacente. On peut aussi améliorer la précision et la fiabilité en produisant un indice de confiance à partir de la triple validation : présence dans UMLS, reconnaissance par l'expression régulière, et multiples occurrences dans le corpus. Enfin, si la fiabilité de cet automate le permet, on peut envisager de supprimer du lexique de l'analyseur les termes reconnus par ces expressions (et éventuellement récurrents). L'allègement du lexique améliorerait les performances et éviterait l'étape de lemmatisation des mots potentiellement source de bruit.

Persistance embarquée, un second niveau de cache

D'un point de vue plus proche de l'implémentation, nous envisageons de mettre en œuvre un second niveau de cache. Certaines procédures génèrent des données intermédiaires qui sont stockées dans le SGBDR à défaut de capacités suffisantes de la mémoire centrale. Dans le cas de l'analyse distributionnelle, ce défaut représente un surcoût important résultant des synchronisations multiples avec le serveur. Des systèmes de persistance embarqués (Apache Derby, Oracle BerkeleyDB, etc.) permettent dans certains cas de décupler les performances en contrepartie du sacrifice de la notion de service, de transaction, de contrainte ou même de langage de requête de haut niveau (comme SQL). Ces moteurs, en gérant les données intermédiaires peuvent rendre nettement plus performante l'intégration, instaurant l'équivalent d'un second niveau de cache, moins performant que la mémoire centrale mais sans contrainte de volume et nettement plus rapide que le SGBDR. Les principaux bénéficiaires seraient l'analyse lexicale et l'analyse distributionnelle dans notre cas. Des stratégies de gestion de ce cache sont alors à prévoir.

Cette amélioration accroît les performances, la portabilité et l'autonomie de chaque composante d'intégration, d'analyse ou d'enrichissement. Mais on peut aussi envisager des bénéfices en termes de qualité de résultat. On peut ainsi décider de désactiver les synchronisations multiples de l'analyse distributionnelle ou la lemmatisation des mots dans l'analyse lexicale. Ce dernier point, rappelons-le, est source d'ambiguïtés, mais augmente le rappel. Des évaluations plus précises doivent mettre en balance les gains et les pertes occasionnés.

Vers une analyse syntaxique

Après les différentes améliorations proposées dans les deux derniers paragraphes, nous souhaitons mettre en œuvre une évaluation comparative et quantitative. Les lacunes des différents analyseurs sont communes dans le contexte des documents biomédicaux : le découpage préliminaire de la phrase à l'aide de la ponctuation détruit la syntaxe de la phrase. Si notre analyseur lexical se confirme à la hauteur de nos attentes, nous l'utiliserons comme prédécoupage, avant l'analyse syntaxique. Nous sommes curieux de mesurer l'impact éventuel de ce prétraitement sur la qualité de l'analyse syntaxique.

Aspects légaux liés à l'intégration de données

Enfin, nous souhaitons soulever une limite inhérente à notre domaine de recherche que nous n'avons que rarement vu aborder : le droit et les licences. UMLS propose quatre types de licences différents avec des droits parfois très restrictifs et le plus souvent ambigus. Pour télécharger PubMed, il faut répondre à un formulaire par écrit et le faire parvenir aux services localisés aux Etats-Unis. Ainsi, quelles sont les conditions liées à une carte, aux données de l'entrepôt. Quelles sont les obligations de l'utilisateur final du système ? Qui a des droits sur les résultats ? Ce problème complexe nécessite pour chaque ressource intégrée de vérifier les

aspects légaux. Qui en serait en charge ? On ne peut pas demander à l'utilisateur d'une carte de remplir de nombreux formulaires, de mener une étude juridique, de sacrifier des droits sur son travail ou encore de télécharger lui-même tous les fichiers dispersés sur la toile. Ce problème est d'autant plus important que l'on souhaite tirer profit de ce type de portail. Les licences des ressources que nous utilisons contiennent des clauses spécifiques à la recherche et aux organisations à but non lucratif. Le problème est plus complexe encore pour des entités commerciales (ou retirant un profit).

CHAPITRE 6

Visualisation, interaction et adaptabilité de la carte

« The power of unaided mind is highly overrated. Without external aids, memory, thought, and reasoning are all constrained. But human intelligence is highly flexible and adaptive, superb at inventing procedures and objects that overcome its own limits. The real powers come from devising external aids that enhance cognitive abilities. »

NORMAN, 1993

6.1 Introduction

Dans le chapitre 3, nous avons argumenté sur l'apport d'une méthode visuelle. Etant donné le volume des données, il est nécessaire de mettre en œuvre des techniques de visualisation adaptables aux différents besoins et aux différents cas d'utilisation. La conception d'une application unique visant à répondre aux différents besoins existants ne nous semble pas réaliste : les besoins sont imprévisibles, et leur inventaire exhaustif impossible. De plus, une telle application serait particulièrement difficile à appréhender pour l'utilisateur.

Nos constats et nos objectifs en matière de visualisation sont les mêmes qu'en matière d'intégration : les approches existantes sont souvent difficiles à mettre en œuvre et à réutiliser car peu extensibles, peu polyvalentes, et peu ouvertes du point de vue des données. Fournir des fonctionnalités de visualisation est pourtant indispensable pour homogénéiser le poste de travail de l'utilisateur, lui permettre d'interpréter de grandes quantités de données, et fédérer le développeur souvent peu initié à ce domaine.

La solution que nous proposons est une boîte à outils graphique. Nous souhaitons mettre à disposition un cadre commun permettant l'unification des interfaces utilisateurs. Cette boîte à outils offre un accès visuel personnalisé aux principales ressources biologiques, en état adapté à un contexte et une application spécifique. Généraliste, elle est avant tout orientée vers la visualisation de données de graphes de notre entrepôt, tout en étant capable de représenter des données multidimensionnelles. Une application, même simple, est généralement liée à des cas d'utilisation variés ; nous mettons l'accent sur l'adaptabilité de l'outil aux différents cas d'utilisations qui se succèdent au fur et à mesure que le biologiste progresse dans son projet.

Ce chapitre se divise en trois sections. Dans la première, nous précisons les besoins et les choix concernant la méthode de visualisation. La deuxième argumente notre choix d'extension d'une boîte à outils graphique existante, Prefuse [Heer, Card et al. 2005], et certaines fonctionnalités que nous avons ajoutées. La dernière section discute l'adaptabilité de l'environnement aux besoins de l'utilisateur qui se présente à plusieurs niveaux :

- l'extraction d'une carte à partir de l'entrepôt de données,
- l'adaptabilité de la visualisation grâce aux fonctionnalités de notre boite à outils graphique,
- la mise en œuvre de critères d'adaptabilité à l'échelle des données et des métadonnées découlant des interactions de l'utilisateur et de ses préférences.

6.2 Choix de visualisation

6.2.1 Les besoins de nos utilisateurs

Contextes

Impliqués dans différents projets avec des biologistes, nous avons recensé plusieurs besoins. Dans le contexte du CEA et du programme ToxNuc-E, certains besoins provenaient de la direction du programme, d'autres directement de chercheurs dans le cadre des sous-projets *Nephrotoxicité* et *Arabidopsis*. Plus récemment, notre collaboration avec Y. Cayre des Hôpitaux de Paris s'est inscrite dans la lignée de travaux menés avec l'Institut Pasteur et du projet *Arabidopsis* du CEA.

Le Programme de recherche « Toxicologie Nucléaire Environnementale » (ToxNuc-E) a pour objectif d'identifier les effets toxiques d'éléments chimiques, radioactifs ou non, utilisés dans la recherche et l'industrie nucléaires. Ces travaux visent à déterminer les mécanismes de toxicité de ces éléments pour l'homme et son environnement et de proposer des procédés de dépollution des sols et de traitement d'éventuelles contaminations. Ce programme inter-organismes, piloté

par le CEA et soutenu par le Ministère de l'Enseignement Supérieur et de la Recherche, fédère depuis 2004 une communauté scientifique pluridisciplinaire d'environ 200 chercheurs/an et 100 doctorants et post-doctorants (CEA, CNRS, Inra, Inserm). Ce programme est divisé en plusieurs sous-projets. Le centre de recherche LGI2P est partenaire de Toxnuc-E pour tout ce qui concerne l'instrumentation technologique de la gestion de programme : aspects sociologiques et techniques du travail collaboratif, analyse et visualisation des interconnections entre sous-projets, entre chercheurs, entre domaines de recherche, gestion de la base des publications (indexation, recherche d'information).

La direction des programmes de ToxNuc-E a soulevé plusieurs besoins en matière de soutien technique. Certaines réponses ont été proposées par l'équipe Cyclone du LGI2P concernant des thématiques de travail collaboratif et d'intelligence collective. Plusieurs besoins visuels ont été explicitement soulignés. Le premier concerne la supervision de programmes : la direction souhaite cartographier les projets, c'est-à-dire visualiser un réseau constitué par les chercheurs, les coordonnateurs, les projets. Toutes ces entités sont reliées en fonction de leur communication et de leur collaboration (figure 6.1). Le second souhait de la direction de programme était de produire une ontologie de la toxicologie nucléaire environnementale et de la visualiser (figure 6.2). En particulier, dans le cadre de la mise en œuvre d'un système de gestion de contenu, cette ontologie peut-être combinée aux documents pour permettre un accès aux documents via une carte de connaissances (figure 6.3).

Toujours au sein du programme ToxNuc-e, nous avons été mis en relation avec des chercheurs de deux sous-projets. Le premier, intitulé « *Nephrotoxicité et Toxico-Cancerogenèse* » évoque des difficultés pour rechercher l'information dans PubMed. Il nécessite une veille bibliographique en lien avec les connaissances sur la cellule. Rappelons qu'au départ de ce projet, Entrez PubMed ne proposait pas encore les mécanismes de personnalisation de MyNCBI. Le second projet concerne l'*Arabidopsis Thaliana* et s'intéresse aux réponses du métabolisme de cette plante suite à des stress par des métaux lourds. Dans ce contexte, les chercheurs n'expriment pas de problèmes en recherche documentaire mais rapportent une difficulté pour accéder et croiser l'information contenue dans les portails du domaine. Les chercheurs passent ainsi de longues heures à copier et coller les données dans leur tableur favori et à les réorganiser pour en obtenir une vue synthétique.

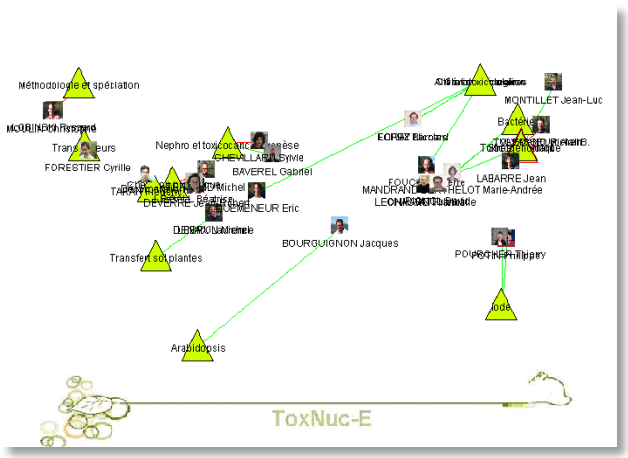


Figure 6.1 – Cartographie des coordonnateurs de projets

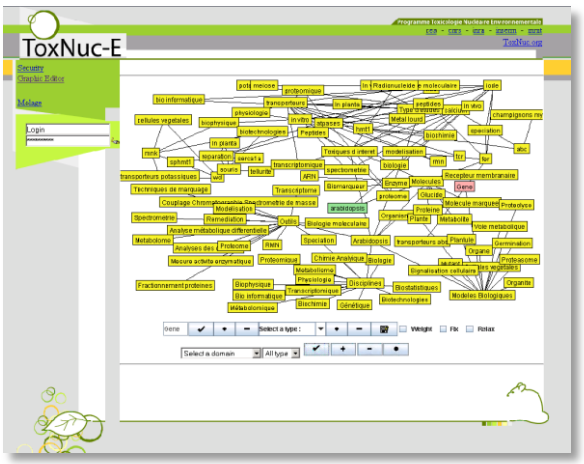


Figure 6.2 – Réseau sémantique (étendu) dans le portail ToxNuc-e

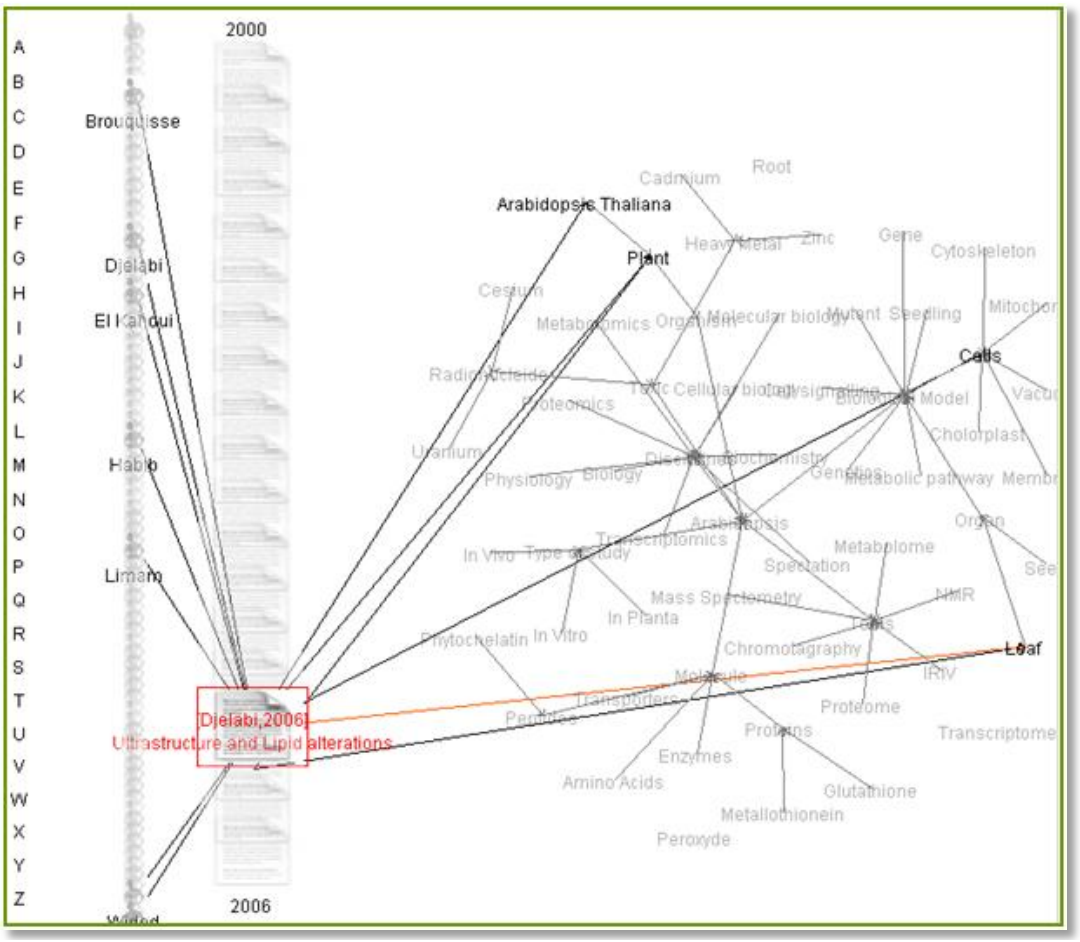


Figure 6.3 – Exploration du contenu documentaire.

Notre collaboration avec l'Institut Pasteur concernaient l'analyse automatisée de données d'expression de gènes (regroupement, classification, etc.) issues de puces à ADN. Les expériences menées par Bozdech et par d'autres sur des puces à ADN ont été réalisée avant le séquençage du génome de Plasmodium Falciparum. L'équipe de l'institut Pasteur a ainsi une expérience à partir de puces à ADN. Une mesure (et ses réplicas) est réalisée pour chacun des quatre principaux cycles érythrocytaires, à l'échelle du génome entier de l'organisme (soit près de 5300 gènes). Les premiers essais ont conduit à des difficultés dues à des problèmes de synchronisation des cellules de l'échantillon n'a pas été efficace. Pour tester nos méthodes et enrichir les données d'expression trop peu fiables, nous avons décidé d'intégrer simultanément des données d'expression externes, publiques. En parallèle, des biomathématiciens ont cherché des méthodes formelles de resynchronisation des données.

Plus récemment, un autre projet nous a conduits à mettre en œuvre des techniques d'analyse de données de puces à ADN. Cette collaboration avec l'université Pierre et Marie Curie et l'hôpital Robert Debré concerne l'étude de l'action de l'acide rétinoïque sur des cellules affectée par la leucémie promyélocytaire aiguë. La plateforme utilisée est produite par Applied BioSystems (AB) et concernent le génome humain complet (soit plus de 30 000 gènes). La qualité des données n'est pas ici problématique. Par contre, durant les années précédentes, les expériences ont été menées par plusieurs biotechniciens de l'unité. Aujourd'hui, on dispose de plusieurs jeux d'expériences qui ne sont pas directement comparables et qui sont individuellement incomplètes. Il est nécessaire de procéder à plusieurs analyses de données et à en intégrer les résultats avec les connaissances du domaine.

Les problèmes rencontrés par les membres de l'institut Pasteur et par l'équipe d'Y. Cayre ne relèvent pas du même ordre : pour les premiers il s'agit d'un problème biologique, pour les seconds, il s'agit des aléas sociaux de la vie du laboratoire. Cependant, du point de vue informatique, les deux équipes de recherche expriment un besoin commun d'intégration de jeux

de données différents et de connaissances du domaine. Ce nouveau projet permet de démontrer les capacités rapides d'adaptation de l'environnement et sa souplesse : les systèmes d'informations sont nombreux, différents des projets précédents. Les données sont par ailleurs plus nombreuses, diversifiées et fiables.

Besoins

Les besoins que nous venons de voir sont variés : la gestion de projet dans le programme ToxNuc-E nécessite de représenter des sous-projets, leurs acteurs principaux (leurs photographies), les interactions sociales et les actes de communication entre eux. Un réseau de termes est visualisé pour la construction d'ontologies. Dans ce cadre, plusieurs ontologies existent déjà, l'ontologie de la toxicologie nucléaire se positionne comme transversale et médiatrice des ressources existantes. L'ontologie obtenue est visualisée comme un réseau de termes pour naviguer dans le système de gestion de contenu. Dans le projet *Arabidopsis*, on souhaite synthétiser des informations fonctionnelles sur des protéines sous forme de tableaux. Dans le projet *néphrotoxicité*, un fond de carte représente l'anatomie de la cellule, on dispose par dessus des connaissances sur la cellule (gènes et annotations) et les documents qui y sont associés. Enfin, dans le cadre des collaborations concernant l'analyse de données d'expression, le besoin est de visualiser des regroupements flous et de comparer ces regroupements. Pour cela, il faut pouvoir superposer plusieurs regroupements, et ajouter des informations fonctionnelles sur les gènes et des références bibliographiques.

Nous n'avons pas été en charge de répondre à tous ces besoins, mais nous avons eu l'opportunité de participer aux discussions relatives à ces différents problèmes. De l'ensemble de ces problématiques, on peut synthétiser globalement plusieurs besoins :

- visualiser des grands graphes, dont les topologies, comme les données, sont hétérogènes (réseaux de termes, de documents, de gènes, etc.),
- conjuguer la visualisation de données multidimensionnelles (données d'expression de gènes ou de protéines) à ces grands graphes,
- visualiser des ensembles et leurs intersections (plusieurs ontologies, comparaisons de regroupements, regroupements flous),
- dessiner un fond de carte (cartographie de la cellule).

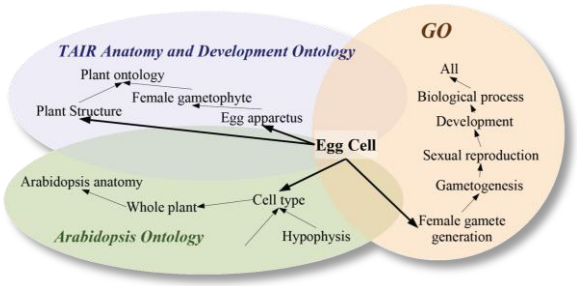
L'entrepôt est structuré sous forme de graphe afin d'être adapté à notre approche visuelle. Cependant, l'extensibilité de notre approche nous interdit toute supposition sur la topologie : l'entrepôt contient des réseaux sociaux, des graphes petits mondes, des réseaux de cooccurrences, des données multidimensionnelles, des regroupements (hiérarchiques, flous, etc.), des ontologies (hiérarchiques, DAG, …), etc. Les caractéristiques de la méthode de dessin de graphe que nous avons choisie sont sa robustesse et sa polyvalence. Les données sont présentes en grande quantité, il est donc préférable de prévoir une visualisation multiéchelle, capable de s'adapter aux différentes tâches. Une méthode permettant une évolution dynamique et animée du graphe est préférable.

6.2.2 Choix d'une méthode de visualisation

Choix d'une technique de visualisation

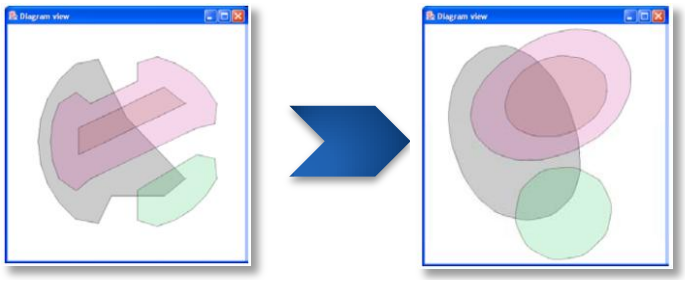
Dans un premier temps, pour disposer les éléments du graphe, nous pensions utiliser un outil de dessin de diagrammes d'Euler (par exemple dans le contexte de la conception d'une ontologie à partir de plusieurs ressources préexistantes, cf. figure 6.4). Les diagrammes d'Euler sont un cas particulier des diagrammes de Venn. Dans les diagrammes de Venn, toutes les zones possibles sont représentées. Dans le diagramme d'Euler, seule une partie des zones est représentée. Ce type de dessin s'avère cependant très difficile à réaliser [Flower and Howse 2002; Flower, Rodgers et al. 2003] (figure 6.5). Les méthodes existantes sont complexes, ne fonctionnent que

pour un nombre réduit d'ensembles, ne sont pas dynamiques, et la *dessinabilité*[1] du diagramme d'Euler est dépendante de la connexité de son graphe dual [Flower and Howse 2002]. De plus, une difficulté s'ajoute lorsque l'on souhaite gérer la surface allouée à chaque zone.



◄ *Figure 6.4 – Exemple d'utilisation des diagrammes d'Euler pour visualiser des ontologiques : chaque ontologie est considérée comme un ensemble de termes (schéma généré manuellement).*

Figure 6.5 – Le dessin des diagrammes d'Euler est complexe. La figure de gauche montre les résultats produits par l'algorithme proposé dans [Flower and Howse 2002], la figure de droite montre le résultat des améliorations esthétiques proposées dans [Flower, Rodgers et al. 2003]. ►



Le graphe correspond à la topologie la plus courante des données qui nous concernent et nous avons porté notre choix sur une méthode de dessin de graphe. Nous avons retenu une approche basée sur un modèle physique de force, reconnue pour sa robustesse [Eades 1984] : si elle ne fournit pas un résultat optimal dans de nombreux cas suivant certains critères (angles, distance, enchevêtrement, etc.), elle offre un résultat satisfaisant quelle que soit la topologie.

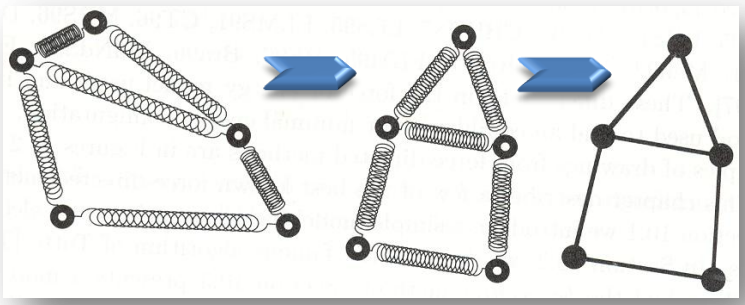


Figure 6.6 (extraite de [Di Battista, Eades et al. 1999]) – Dessin de graphe à base de force : on associe à chaque arête un ressort. On simule alors l'exercice des forces des ressorts, et on obtient une disposition stable (un minimum local).

Le principe de fonctionnement de l'algorithme est assez intuitif. On positionne les nœuds du graphe, puis à chaque arête on associe un ressort (figure 6.6). Lorsque la distance entre les nœuds est plus importante que la longueur au repos du ressort, alors le ressort exerce une force pour rapprocher les nœuds. Au contraire, lorsque le ressort est comprimé, il exerce une force de répulsion. On simule alors les forces exercées jusqu'à obtenir un graphe qui se stabilise ; c'est-à-dire jusqu'à l'obtention d'un minimum local dans la fonction correspondant au stress cumulé de tous les ressorts.

Cette méthode possède de nombreux avantages : elle est intuitive, prévisible, dynamique et paramétrable. Intuitive, son fonctionnement est facile à comprendre pour l'utilisateur. S'il ajoute ou retire un élément (nœud ou arête), la visualisation ne va pas être totalement bouleversée mais va simplement évoluer de façon continue. Il peut ainsi prévoir l'effet de ses actions sur le

[1] *drawability*

dessin du graphe. Enfin, en modifiant le poids des sommets et les longueurs et constantes d'élasticité des ressorts, on peut produire de multiples vues et les alterner de façon continue.

6.2.3 Evaluation de la méthode de visualisation

Nous avons précisé que nous souhaitons pouvoir visualiser des données multidimensionnelles (cf. 4.2.1 page 120). Une méthode courante pour la réduction à deux dimensions d'un espace vectoriel consiste à générer un graphe complet à partir de la matrice de distances. La longueur des ressorts est alors obtenue à partir de la distance dans l'espace vectoriel d'origine [Saporta 2006]. Cette méthode[1] n'étant pas initialement conçue pour la projection de données multidimensionnelles, nous avons réalisé une évaluation rapide de sa qualité. Pour cela, nous l'avons comparée avec une analyse en composantes principales (ACP), sur un échantillon de données d'expression, en nous intéressant en particulier au critère de conservation des distances. Dans le cadre de MDS, nous avons réalisés plusieurs tests dont nous avons extrait la moyenne afin de réduire l'influence des minima locaux.

Critères d'évaluation[2]

Nous avons utilisé plusieurs mesures afin d'évaluer la qualité de la projection. Dans un premier temps, nous avons mesuré la conservation globale de la distance. Nous avons linéarisé de façon ordonnée toutes les distances correspondant aux couples de nœuds possibles. Ceci a été calculé dans l'espace original et dans le plan projeté. Nous avons alors comparé la position de chaque couple dans le tableau stockant les données de l'espace initial, et dans le tableau de l'espace final. Puis nous avons calculé la somme des écarts observés. Plus la valeur est grande, moins la projection respecte les distances. Nous nommons cette mesure *« désordre de distances »* (figure 6.7).

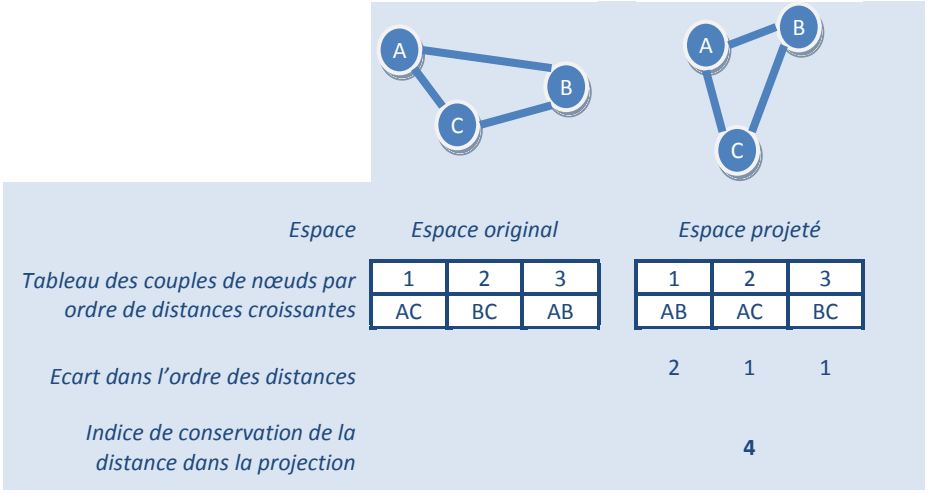


Figure 6.7 – Exemple de calcul du « désordre de distances » *introduit par une projection.*

Afin d'évaluer la conservation des angles, nous avons effectué la mesure non plus en considérant les distances, mais les aires des triangles. Enfin, nous avons considéré une autre mesure qui se focalise non plus sur la distance, mais sur le voisinage. Cet indice représente le pourcentage de conservation des *n* plus proches voisins de l'espace initial dans l'espace projeté. Cette mesure nous intéresse, car dans un espace complexe de données, l'utilisateur ne mesure

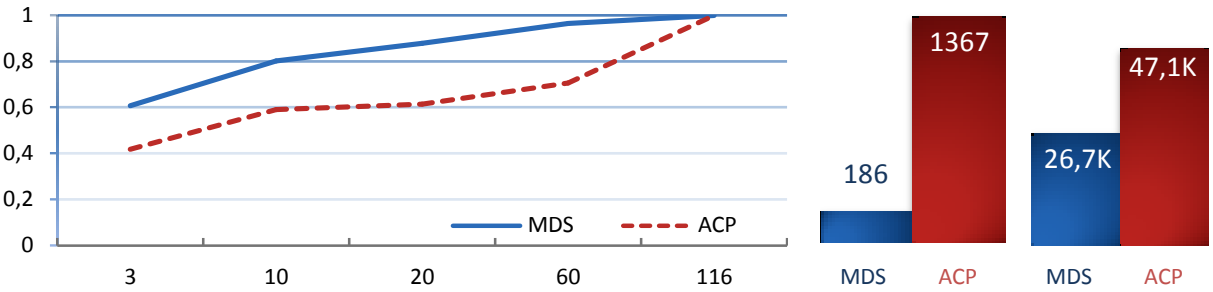
[1] Par la suite, nous utilisons l'acronyme anglo-saxon pour désigner cette méthode : MDS (multidimensional scaling).

[2] Ce travail a été réalisé dans un cadre d'un co-encadrement avec Reena Shetty d'une mini-mission mathématique. Je remercie Gérard Dray, Stefan Janaqi et Jacky Montmain pour leurs conseils concernant les aspects mathématiques.

pas la distance entre les objets, mais associe simplement les objets les plus proches [Mackinlay 1986]. Ce sont ces critères qui sont représentés dans la suite.

MDS *versus* ACP

L'ACP est certainement une des méthodes d'analyse de données multidimensionnelles les plus utilisées et considérée comme satisfaisante. Nous avons donc comparé le MDS avec cette méthode qui nous sert de référence (figure 6.8). Les résultats montrent que le MDS, quelle que soit la mesure de qualité, s'avère plus respectueuse des distances, des aires, et du voisinage. On peut expliquer ce résultat du fait que l'ACP ne traite que les deux premiers vecteurs propres de l'espaces. MDS au contraire prend en compte l'ensemble des vecteurs dans le dessin.



Pourcentage de conservation du voisinage (3, 10, 20 et 60 plus proches voisins).

Indice de *désordre de distances*

Indice de désordre des aires

Figure 6.8 – Comparaison de la qualité des résultats fournis par MDS et ACP. L'échantillon correspond aux données d'expression [Bozdech, Llinás et al. 2003] : 116 gènes pour 46 dimensions.

MDS : complexité

La principale limite du MDS est sa complexité. La mesure de qualité précédente est réalisée en générant un graphe complet où chaque longueur de ressort correspond à la distance euclidienne dans l'espace vectoriel d'origine. Ceci implique la génération de n^2 arêtes pour n vecteurs, et la complexité totale de l'algorithme pour i itérations est de $O(n^2 \times i)$. Cette complexité est importante, et nous avons recherché :

- dans quelle mesure il est possible de réduire le nombre d'arêtes générées sans réduire fortement la qualité de la projection ?
- quelle stratégie adopter pour supprimer les arêtes ?

Nous comparons trois méthodes (figure 6.9) : on génère les arêtes correspondant aux n plus proches voisins, aux n voisins les plus éloignés, ou de façon aléatoire.

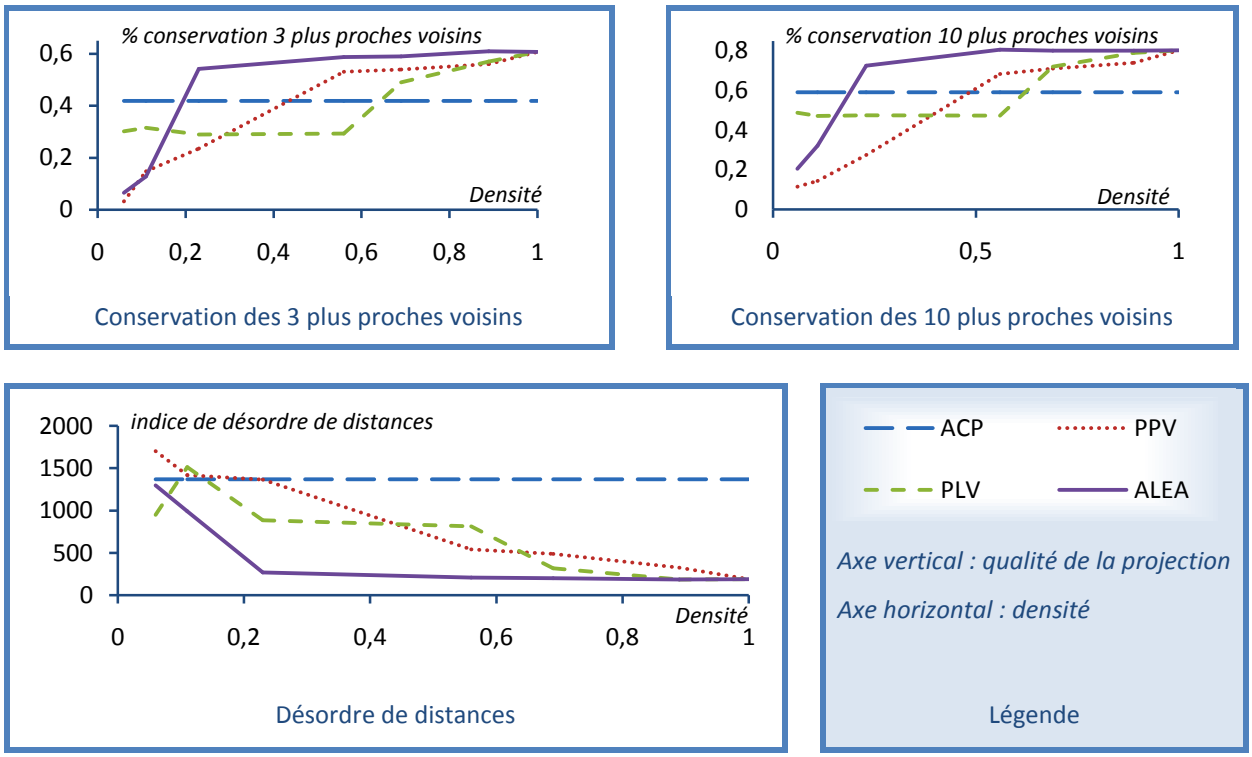
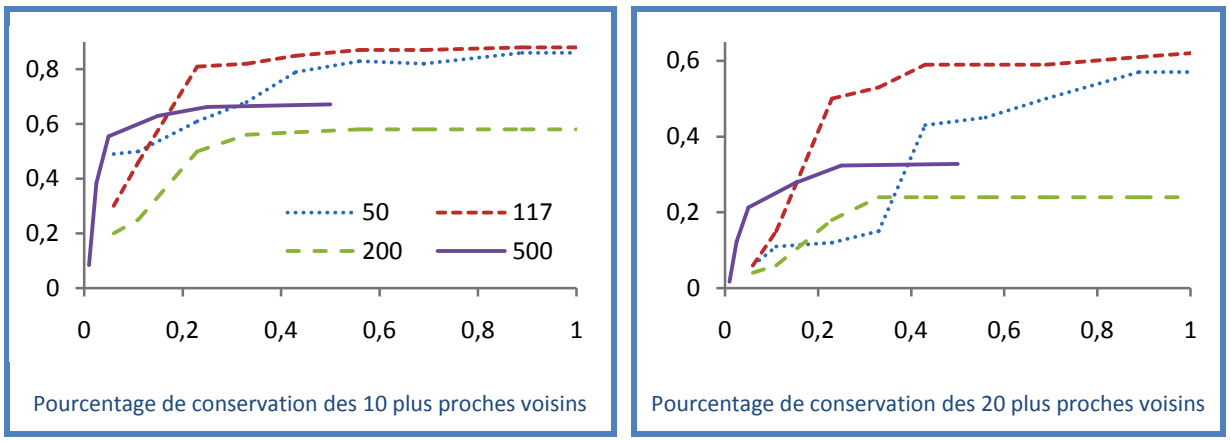


Figure 6.9 – Chaque graphe permet d'évaluer la méthode de construction du graphe en MDS pour un critère d'évaluation donné : 3 et 10 plus proches voisins, désordre de distances. L'axe horizontal décrit la densité du graphe. Chaque courbe indique la qualité de la projection suivant une méthode de génération du graphe de distances différente : plus proches voisins (PPV), plus lointains voisins (PLV), et suppression aléatoire des arêtes. Nous avons par ailleurs rappelé la qualité produite par l'ACP pour chaque méthode d'évaluation (ligne horizontale bleue). La légende est commune aux trois graphiques.

La figure précédente montre les résultats de ces évaluations. Nous concluons que la méthode la plus adaptée pour générer le graphe de distances consiste à filtrer aléatoirement les arêtes. Dans l'expérience précédente, on constate que la qualité se dégrade fortement lorsque la densité est inférieure à 25%. Dans quelle mesure ce point varie-t-il en fonction de la taille des données ? L'expérience suivante mesure l'évolution de la qualité (axe vertical) en fonction de la densité du graphe (horizontale), suivant la méthode aléatoire de filtrage d'arêtes (figure 6.10).



Axe vertical : qualité de la projection - Axe horizontal : densité

Figure 6.10 – Evolution du compromis densité/qualité en fonction du nombre de vecteurs.

A partir de ces expériences, nous établissons un premier constat qui n'est pas traduit dans ces diagrammes : plus le graphe est petit (faible nombre de nœuds), plus les résultats sont variables. Plus le graphe est grand (le nombre de nœuds est important), plus les résultats sont répétés et moins les minimas locaux introduisent de variations. Lorsque le graphe est petit, la visualisation

est plus facile, et la complexité réduite : nous nous focalisons sur les grands graphes. Nous avons identifié plusieurs caractéristiques récurrentes :

- plus le graphe est grand, plus le rapport densité/qualité optimal semble se réduire,
- nous parvenons à afficher un espace multidimensionnel de 500 éléments avec une qualité raisonnable, mais avec une visualisation saccadée (de l'ordre de 1 à 2 images par secondes),
- à partir de 100 vecteurs approximativement, on peut supprimer 75% des arêtes avec une perte de qualité mineure ; en dessous, il n'est pas nécessaire de réaliser de telles optimisations.

6.2.4 Bilan concernant la visualisation

La méthode de dessin de graphe que nous avons mise en œuvre répond à l'essentiel de nos attentes. Elle permet de visualiser de nombreuses topologies de graphes de façon satisfaisante tout en étant intuitive pour l'utilisateur. Dynamique, elle est adaptée à une problématique interactive : filtrage ou ajout d'éléments dans le graphe en cours d'utilisation, possibilité d'adaptation successive et continue à plusieurs scenarios d'utilisation. En jouant sur les constantes d'élasticité, les poids des nœuds et la longueur à vide des ressorts, il est simple de combiner des données hétérogènes et de les pondérer. On peut éventuellement en laisser le soin à l'utilisateur. L'inconvénient est que ce paramétrage reste expérimental et que nous ne disposons d'aucune méthodologie permettant d'aboutir de façon guidée et certaine à un résultat satisfaisant.

La projection multidimensionnelle fournit une qualité de résultats tout à fait satisfaisante, compétitive par rapport à une technique de référence : l'ACP. Du point de vue des performances, la réduction de la densité du graphe nous permet de visualiser jusqu'à 500 vecteurs. Ces expériences ont été menées sur des données d'expression, il faudrait confirmer ces résultats sur d'autres échantillons de données.

6.3 Mise en œuvre

6.3.1 Prefuse

La section précédente a présenté la méthode de visualisation sur laquelle repose notre environnement de cartographie. Il ne s'agit pas ici de mettre en œuvre une application, mais une boîte à outils permettant la construction d'applications graphiques riches, reliées à l'entrepôt ou à une carte. Pour cela, nous avons repris et étendu une boîte à outils existante : Prefuse (version alpha) [Heer, Card et al. 2005]. Cette boîte à outils possède trois avantages majeurs :

- elle est implémentée de façon performante (double buffering [Flanagan 1999], gestion des forces par l'algorithme de Runge-Kutta d'ordre 4 [Abramowitz and Stegun 1972], utilisation d'un algorithme performant pour les forces de répulsion des éléments trop proches [Barnes and Hut 1986], etc.),
- elle repose sur une architecture robuste et souple (figure 6.11 et figure 6.12) [Card, Mackinlay et al. 1999],
- l'auteur a réalisé une évaluation de sa boîte à outils et montré qu'un développeur pouvait en moins d'une heure aboutir à un résultat, l'approche la plus efficace étant de réutiliser un code existant.

Architecture générale

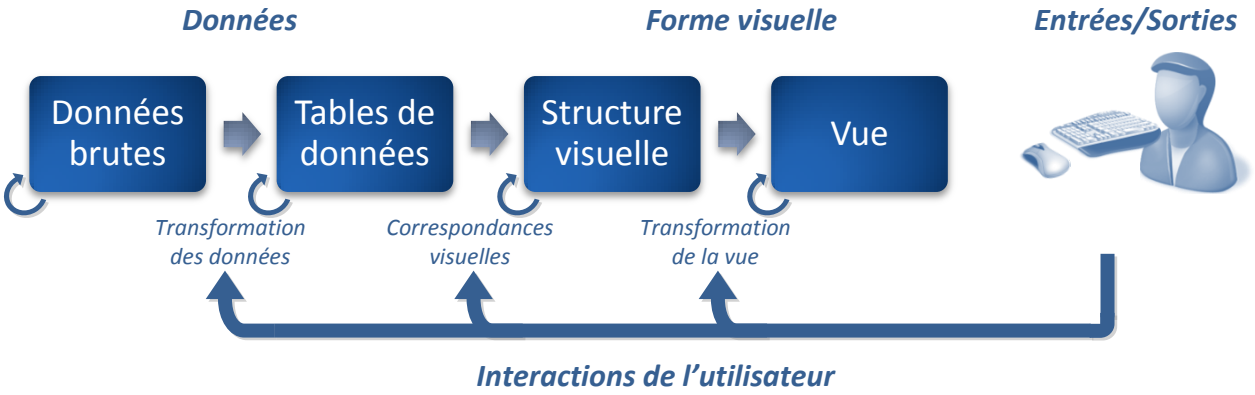


Figure 6.11 – Modèle à état de [Card, Mackinlay et al. 1999] et [Chi and Riedl 1998]

Cette architecture dénommée « Data State Model » (modèle à état de données) a été introduite par Ed H. Chi [Chi and Riedl 1998] à partir du modèle de Stuart P. Card décrit dans [Card, Mackinlay et al. 1999]. Ed Chi montre que ce modèle est équivalent, du point de vue de l'expressivité, à son principal rival, le modèle à base de flux de données [Chi 2002]. Notons que ce modèle a été étendu par la suite par M. Carpendale qui distingue la présentation de la représentation [Carpendale 1999]. Cela rejoint l'approche de Christophe Tricot dans le contexte de la cartographie sémantique [Tricot 2006].

L'implémentation de ce modèle dans Prefuse est illustrée dans la figure 6.12. Les données brutes sont stockées dans un modèle de graphe (appelé modèle abstrait). Après l'application d'un filtre, on ne conserve que les données à visualiser dans une nouvelle structure de données (modèle visuel). Vient après le calcul du rendu à l'écran de ces données visuelles : chaque objet est associé à un « *Renderer* » qui détermine la façon dont il est dessiné. Cette architecture respecte donc les principes de la séparation fonctionnelle des IHM. Les interactions de l'utilisateur sont finalement retransmises à différents niveaux de cette architecture, en fonction du besoin. Par exemple :

- lorsque l'utilisateur zoome ou se déplace dans l'écran, c'est au niveau du rendu visuel qu'est transmis l'événement,
- lorsqu'il masque un élément, le sélectionne, le fige, etc. c'est au niveau de la forme visuelle qu'intervient l'action,
- lorsque l'utilisateur importe des données, supprime un nœud, affecte un nouveau nom, etc., c'est dans le modèle abstrait qu'est répercutée cette action.

Ces actions ne sont que des exemples, de nombreuses autres interactions sont possibles que nous ne détaillons pas ici.

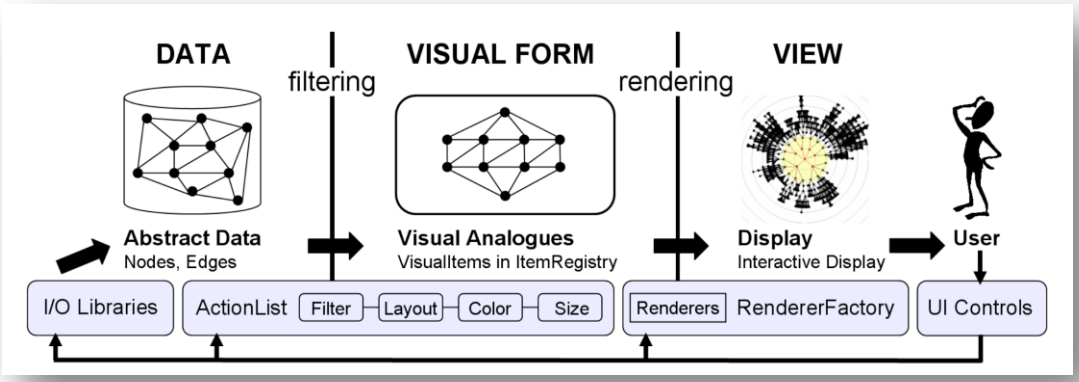


Figure 6.12 – Implémentation de modèle à état dans Prefuse [Heer, Card et al. 2005].

Fonctionnalités existantes

Prefuse propose de nombreuses fonctionnalités dans sa version *alpha*. Cette boîte à outils met à disposition différentes techniques de visualisation, autres que le modèle de forces : dessins hyperboliques, polaires et radiaux de graphes, dessins horizontaux d'arbres ou par des TreeMaps [Shneiderman 1992], *plotting*, etc. De plus, il met à disposition des techniques de distorsions (fish-eye et bifocal) que l'on peut mettre en œuvre notamment dans des menus [Bederson 2000]. Enfin, il est possible de combiner ces distorsions et ces visualisations : on peut dessiner plusieurs vues de façon synchrone, leur ajouter ou non des distorsions, ajouter une force évitant la superposition de deux étiquettes à la visualisation radiale, etc.

La figure 6.13 présente plus en détail un exemple de visualisation de forces. Il est possible d'obtenir une vue synchrone de taille réduite, qui joue le rôle d'un panneau d'aperçu global. De nombreuses interactions sont paramétrables :

- vues multiples et synchrones (panneau d'aperçu),
- l'élément cliqué devient rouge,
- le voisinage direct est colorié en orange,
- en tirant un nœud (glisser-déposer) il est possible de manipuler le graphe,
- on peut zoomer et se déplacer à volonté dans le graphe
- un panneau d'information donne les informations sur la taille des données et la fréquence de rafraîchissement,
- il est possible d'extraire des captures d'écran hautes définition.

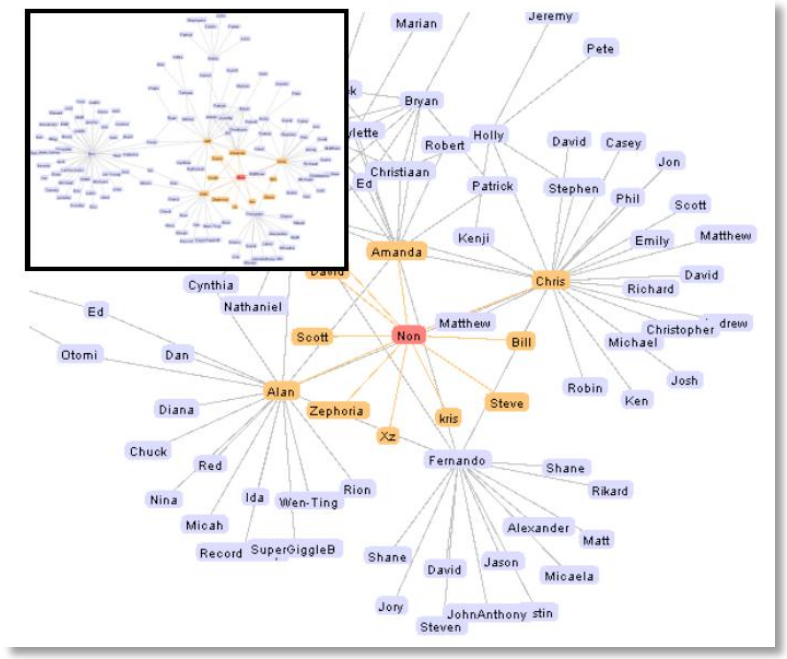


Figure 6.13 – La visualisation à base de forces bénéficie de nombreuses fonctionnalités : panneau d'aperçu, lentille logique (le sommet cliqué devient rouge, ces voisins directs deviennent oranges). Il est possible de se déplacer dans le plan et de zoomer. Il est possible d'obtenir des captures d'écran avec une excellente définition (limitée uniquement par la mémoire disponible pour la machine virtuelle).

Développement simple

Une motivation importante dans notre choix de cette boîte à outils provient de son architecture et de sa simplicité : l'utilisateur peut construire une interface graphique en moins d'une heure. Dans le contexte de la cartographie des connaissances biologiques, ceci est essentiel : la plupart du temps, ni le développeur, ni l'utilisateur n'est familier du domaine de la visualisation. Il faut donc proposer une visualisation robuste, polyvalente et performante, mais il faut aussi que la conception de cette visualisation reste simple.

L'évaluation réalisée par J. Heer et notre expérience de cette boite à outils confirme ce choix. Pour générer une visualisation radiale, il faut 23 lignes, l'ajout de force nécessite 7 lignes de code. Colorier le voisinage nécessite 2 lignes tout comme un *fish-eye*. Enfin, 7 lignes permettent l'ajout d'un panneau d'aperçu, un déplacement et un zoom. Tout ceci est réalisé par de simples copier/coller de code.

6.3.2 Extension des fonctionnalités

Cependant, malgré les qualités qui nous ont fait choisir cette librairie, si les fonctionnalités présentes sont très utiles, elles restent insuffisantes. Nous avons été amenés à modifier certaines parties centrales de la boîte à outils et à ajouter des fonctionnalités (figure 6.14). Ces modifications portent par exemple sur :

- la suppression de l'inertie dans l'intégrateur de Runge-Kutta,
- la prise en compte de la masse individuellement pour chaque nœud, de la constante d'étirement et de la longueur à vide pour chaque arête,
- la modification du modèle pour l'accès aux données de l'entrepôt,
- l'optimisation des performances,
- l'affichage d'info-bulles contextuelles au survol de la souris.

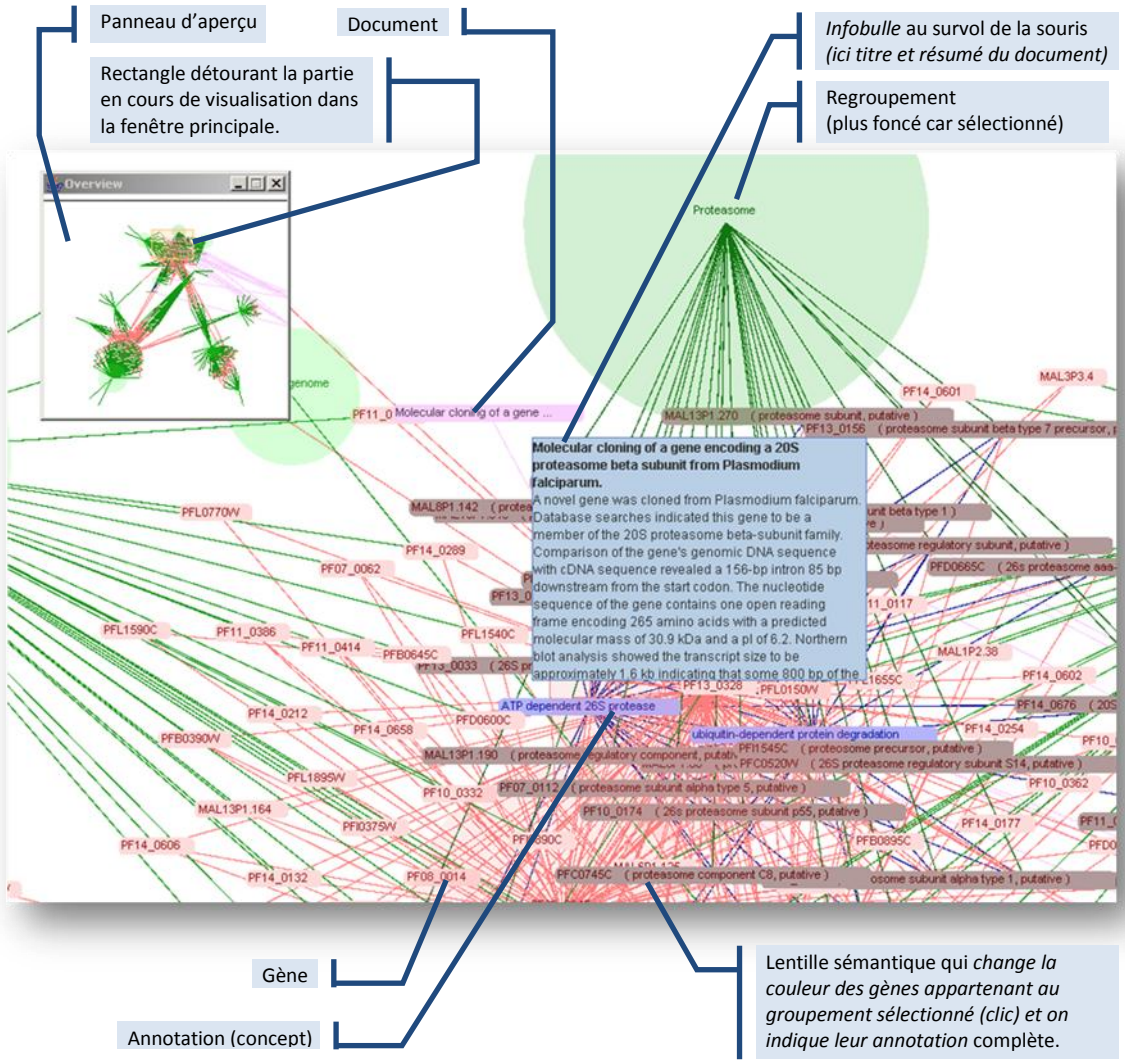


Figure 6.14 – Capture de notre environnement dans le contexte de l'analyse de données d'expression. Cette capture est volontairement chargée du point de vue des données, l'objectif étant de donner un aperçu rapide des fonctionnalités.

Dans cette section, nous ne détaillons pas ces modifications et nous nous concentrons sur l'ajout de certaines fonctionnalités. Nous souhaitons en effet montrer l'extensibilité de la boîte à outils, et la simplicité de développement : l'utilisateur peut décrire des fonctions avancées en quelques lignes. On peut même envisager une spécification sous forme de script ou personnalisable interactivement par l'utilisateur. Les extensions que nous développons dans la suite de ce mémoire sont :

- la gestion des types,
- la gestion des vues multiples,
- le retour utilisateur,
- la gestion de sélections,
- la gestion de lentilles logiques.

6.3.2.1 *Evolutions mineures diverses*

Les extensions que nous décrivons dans les sous-sections suivantes (6.3.2.2 à 6.3.2.5) sont importantes. Ici, nous souhaitons dans un premier temps décrire quelques fonctionnalités mineures et simples à développer afin de montrer dans quelle mesure l'architecture précédente est souple et extensible. L'environnement actuel est expérimental ; pour les versions futures, de nombreuses fonctionnalités sont envisagées afin de répondre à des besoins fréquents et variés,

dans la perspective d'accélérer le développement d'interfaces utilisateurs et de proposer toujours plus de services pour rendre cet environnement attractif pour le programmeur.

Les scripts qui suivent proviennent d'extensions que nous avons développées. Ils sont actuellement écrits en langage Java, directement dans le code de l'application cliente. On peut envisager des les isoler dans du code XML en utilisant des mécanisme d'injection de dépendance ou en proposant à termes un langage de script dédié.

Retour utilisateur dans le glisser-déposer & gestion des vues multiples

Dans la figure 6.14, on remarque la présence d'un panneau d'aperçu. Ce panneau constitue une seconde vue complète dans laquelle il est possible de zoomer, de se déplacer, etc. La différence entre les deux fenêtres (principale et aperçu) provient des dimensions et de la gestion des interactions : les clics sur la fenêtre d'aperçu permettent de recadrer la fenêtre principale, de la déplacer, etc. De la même façon, dans la fenêtre principale il est possible de recadrer la vue. En l'état, la boîte à outils ne dispose pas d'un retour utilisateur concernant la synchronisation des deux vues et de l'interaction simultanée sur ces deux vues.

Pour sélectionner des éléments, ou pour recadrer la vue (zoom + déplacement), l'utilisateur n'a qu'à réaliser un glisser/déposer[1] avec la souris en maintenant la touche « maj » ou « ctrl » enfoncée. Pendant cette sélection, un rectangle transparent délimite la région choisie par l'utilisateur dans la fenêtre principale et dans la fenêtre d'aperçu, où que soit réalisé le glisser-déposer. Pour ce faire, nous avons simplement introduit un nœud rectangulaire dans les données visuelles, les interactions de l'utilisateur mettant à jour la visibilité et les coordonnées de cet objet.

Au final, nous avons introduit une dizaine de classes comportant chacune peu de code. Pour le développeur, il suffit de copier-coller 3 lignes de code dans la classe principale de l'application pour accéder à ses fonctionnalités.

Eviter la superposition de regroupements de même types

Une limite que nous avons rencontrée dans notre visualisation concerne les regroupements de même type qui se chevauchaient. Dans le contexte de *Plasmodium Falciparum* et de nos expérimentations sur le jeu de données de Bozdech [Bozdech, Llinás et al. 2003], nous avons souhaité instaurer une distance entre deux clusters flous et entre deux catégories définies par Bozdech. Cette distance est calculée par la somme des rayons des deux nœuds. Pour ce faire, nous avons ajouté dans le graphe des données des arêtes entre les nœuds de regroupement. Ces arêtes ont un rôle répulsif et une élasticité presque nulle. Cette solution est simple à mettre en œuvre et offre un avantage : les nœuds de séparation ont leur propre type, on peut donc activer ou désactiver cette contrainte au travers du panneau de contrôle (`ClusterSeparatingEdges`, figure 6.16). Nous abordons cette notion de type dans la section suivante.

Menus contextuels

Tout l'intérêt de l'interface est l'accès contextuel à une information : le survol d'un concept affiche sa définition dans une *infobulle*, pour un document c'est son titre complet et son résumé qui sont affichés. Différentes interactions doivent être contextuelles : supprimer, masquer, figer des éléments. Nous avons donc introduit des menus contextuels (Swing). La conception de ces menus a été réalisée, et l'ajout pour le développeur ne prend désormais que deux lignes de code. De plus, il permet d'interroger des portails distants : un fichier XML décrit sous forme de préfixe et suffixe la syntaxe d'interrogation d'un portail *via* HTTP. Dès lors, il est possible, en configurant deux lignes de XML, d'ajouter pour un ou plusieurs types de nœuds un nouveau portail (figure 6.15). L'utilisateur à partir d'un clic droit peut ouvrir un document dans PubMed afin d'accéder au contenu intégral, etc.

[1] *Drag & drop*

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<ROOT>
<SEARCH-ENGINE type="gene">
  <SET name="Gene DBs" alt="Gene relative databases">
     <SITE name="Entrez Gene" alt="gene-centered information" id="2"
        prefix="http://www.ncbi.nlm.nih.gov/gquery/gquery.fcgi?db=gene&amp;cmd=search&amp;term="
        suffix="" />
     <SITE name="PlasmoDB" alt="The Plasmodium Genome Resource" id="3"
        prefix="http://www.plasmodb.org/plasmo/processQuestionSetsFlat.do?question...locus_tag%29="
        suffix="&#38;questionSubmit=Get+Answer&#38;go.x=16&#38;go.y=10&#38;go=go"  />
  </SET>
  <SITE name="PubMed" alt="biomedical literature citations and abstracts" id="1"
     prefix="http://www.ncbi.nlm.nih.gov/gquery/gquery.fcgi?db=PubMed&amp;cmd=search&amp;term="
     suffix="" />
</SEARCH-ENGINE>
<SEARCH-ENGINE type="default">
  <SITE name="Entrez Gene" alt="gene-centered information" id="2"
     prefix="http://www.ncbi.nlm.nih.gov/gquery/gquery.fcgi?db=gene&amp;cmd=search&amp;term="
     suffix="" />
  <SITE name="PlasmoDB" alt="PlasmoDB" id="3"
     prefix="http://www.plasmodb.org/plasmo/processQuestionSetsFlat.do?question...locus_tag%29="
     suffix="&#38;questionSubmit=Get+Answer&#38;go.x=16&#38;go.y=10&#38;go=go" />
  <SITE name="PubMed" alt="biomedical literature citations and abstracts" id="1"
     prefix="http://www.ncbi.nlm.nih.gov/gquery/gquery.fcgi?db=PubMed&amp;cmd=search&amp;term="
     suffix="" />
</SEARCH-ENGINE>
</ROOT>
```

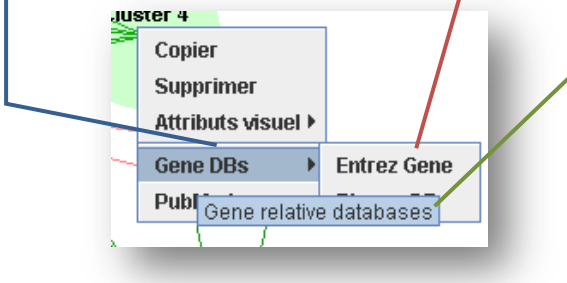


Figure 6.15 – Un fichier XML permet de configurer le menu contextuel. Quelques minutes suffisent pour ajouter un nouveau portail à la liste et le restreindre si besoin est à un type particulier d'éléments. A l'avenir, il faut ajouter la possibilité de spécifier l'accession adaptée au portail et à la source.

6.3.2.2 Gestion des types

La boîte à outils est adaptée à la manipulation des données de l'entrepôt. Ainsi chaque nœud et chaque arête du graphe est liée à la donnée de l'entrepôt au travers de l'API Objet. Les données relatives à chaque nœud et arêtes peuvent être automatiquement chargées à l'initialisation de l'application. L'héritage proposé dans l'API objet permet d'étendre les classes nœuds et arêtes afin de leur ajouter des fonctionnalités (analyse de séquence, …).

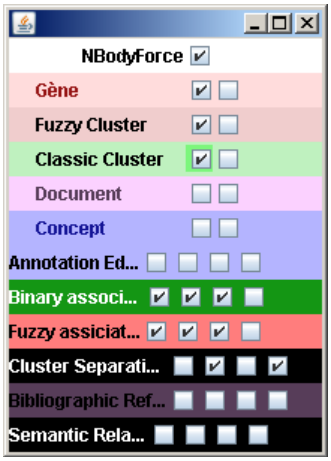
Cet affinement du typage peut être utilisé par une application cliente. Il n'est cependant ni prévisible, ni imposé à l'application cliente. Lors de la visualisation, un typage s'avère important : on doit pouvoir distinguer un gène d'une protéine, d'un document et d'un concept. Mais à nouveau les besoins ne sont pas prévisibles : dans une recherche d'information, il faut distinguer le résumé de PubMed d'une synthèse d'OMIM. Dans une analyse de données biologiques, on peut souhaiter distinguer un ADN d'un ARN, d'un ADNc, etc. Durant la conception d'une ontologie, on souhaite pouvoir distinguer des types sémantiques de concepts, ou les relations *« est un »* et *« partie de ».*

Le développeur a ainsi la possibilité de typer les objets visualisés de façon indépendante du typage de l'API objet et de l'entrepôt. Il peut exploiter simultanément ces trois typages. L'interface `GraphInitializer` permet de gérer l'instanciation des types d'objets et leur initialisation. L'interface `TypeMapper` permet de définir les critères d'attribution d'un type visuel (valeur de la relation sémantique par exemple). Dans les deux cas, nous recommandons d'étendre les implémentations par défaut de ces deux interfaces.

Cette gestion du typage est utilisée durant toute la visualisation : il est possible d'affecter un rendu spécifique à un type, de limiter la sélection d'un ensemble d'éléments à un type donné, de

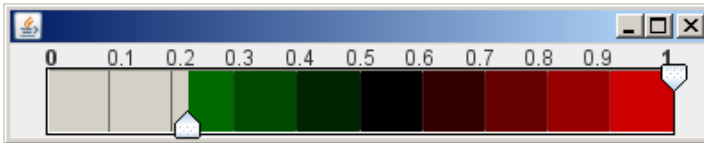
paramétrer les lentilles en fonction de types, ou encore, de manipuler les éléments visuels de façon globale (figer ou masquer un type, cf. figure 6.16), etc.

Un « contrôleur » a été conçu et mis en place dans le cadre de nos expérimentations. Il est essentiellement destiné au développeur, pour lui permettre de trouver un paramétrage satisfaisant la vue d'un opérateur humain. La visualisation étant intuitive, on peut cependant envisager d'étendre son usage à l'utilisateur avancé. La première ligne de cette fenêtre permet d'activer la force de répulsion de tous les éléments, afin d'éviter leur recouvrement. Chaque ligne suivante est associée à un type visuel. Les cinq premières correspondent à un type de nœud : la première case à cocher permet de rendre visible les nœuds de ce type, la seconde permet de les figer. Toutes les lignes qui suivent (au nombre de 6) correspondent à des types d'arêtes. La première case permet de les rendre visibles, la seconde de rendre les forces actives. Les deux dernières cases permettent de n'activer une force qu'en attraction ou en répulsion.



◀ *Figure 6.16 – « Contrôleur » permettant de paramétrer la visualisation dynamiquement et par rapport aux types visuels.*

▼ *Figure 6.17 – Le «* BiSlider *» permet de gérer des intervalles de valeur. Ici, nous l'utilisons pour filtrer le degré d'appartenance dans le regroupement flou.*



Il est possible de filtrer les éléments d'un type sur un intervalle de valeur. Nous utilisons pour cela un « BiSlider », qui permet d'afficher une distribution et de gérer une valeur minimum et maximum au sein d'un même composant graphique. L'usage spécifique présenté dans la figure 6.17 concerne le filtrage des arêtes générées par le regroupement flou. Un contrôleur étendu permet de façon générique de gérer de tels intervalles pour chaque type. Ce contrôleur étendu permet aussi de gérer des valeurs par défaut de masse pour les nœuds, d'élasticité et de longueur pour les arêtes.

Outre la mise à disposition d'outils expérimentaux pour concevoir la carte, ces outils montrent que cette gestion des types est totalement dynamique : elle peut être utilisée au travers de nombreuses fonctionnalités, et dans différentes composantes de l'architecture générale : le BiSlider de la figure 6.17 intervient dans le filtre, l'activation des forces dans le simulateur (action), la visibilité dans le rendu, etc. La manipulation de ces contrôleurs s'est cependant avérée complexe pour l'utilisateur.

6.3.2.3 Lentilles : sélections, filtres et modifieurs

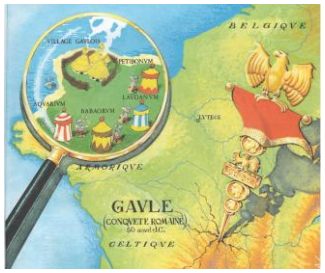
Les lentilles optiques, dans notre quotidien, nous permettent de discerner des détails que nous n'arrivons pas à voir autrement. En visualisation d'information, une approche courante dénommée *« focus + context »* consiste à mettre en évidence l'information au centre de l'intérêt de l'utilisateur, en conservant une vue macroscopique du contexte entourant cette information. C'est par exemple le rôle de la fenêtre d'aperçu présentée précédemment.

Certaines techniques informatiques sont appelées lentilles car elles permettent de focaliser l'attention de l'utilisateur sur certaines parties de l'information de la carte (figure 6.18). La plus élémentaire est la loupe, qui a le désagrément de masquer l'information immédiatement autour de la région d'intérêt. Pour résoudre ce problème, des distorsions ont été mises en œuvre comme le *fish-eye [Sarkar and Brown 1992; Carpendale 1999]*. La limite de cette approche est que le texte en périphérie de la lentille est visible, mais déformé et illisible. Il existe alors d'autres techniques interactives permettant de mieux gérer le focus et contexte : la géométrie

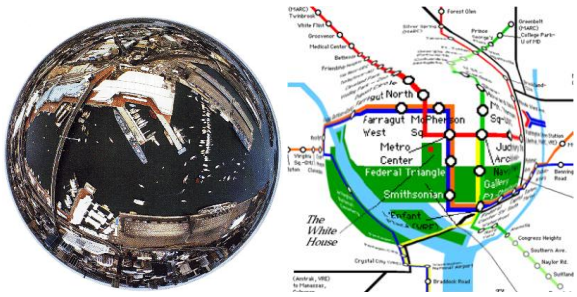
hyperbolique qui se révèle difficile à manipuler [Munzner 2000] et la visualisation polaire qui réduit la charge cognitive pour l'utilisateur [Tricot 2006].

En 1981, G.W. Furnas a proposé une *distorsion logique* (ou discrète) [Furnas 1981]: le texte n'est pas déformé, mais plus une information est éloignée du texte édité (du curseur), plus il est grossier/simplifié. Cette étude se fait dans le cadre de la programmation en C et de la rédaction de documents longs. On ne voit alors apparaître autour du paragraphe en cours d'édition que le plan dans un traitement de texte ou les prototypes de fonction dans un code source. Cette approche évolue alors plus tard en « lentilles magiques » (*magic lenses*) permettant des distorsions logiques sur une région géométrique définie [Bier, Stone et al. 1993] : afficher une information supplémentaire, changer de couleur, etc.

Avec Prefuse, nous avons expérimenté les distorsions optiques pour visualiser le graphe. Comme les Magic Lenses, elles proposent une information détaillée sur une petite région de l'écran. Dans le contexte d'un grand graphe visualisé à l'aide de forces, les résultats sont insatisfaisants. Le problème n'est pas de détailler la région qui entoure les nœuds, mais de mettre en évidence des éléments d'informations en lien avec un centre d'intérêt, dans un contexte donné. Ce sont les lentilles métier que nous proposons. Cette approche répond à un besoin similaire de celui abordé par les systèmes à base de liens [Cohen-Boulakia, Davidson et al. 2006; Durand, Labarre et al. 2006] : la lentille correspond à un motif de chemin. La différence est que la lentille applique de façon contextuelle à un objet. Le chemin permet d'appliquer des filtres de sélection plus complexes, mais la lentille est interactive et s'utilise plus simplement. Rappelons que la présence des lentilles n'exclut pas la spécification de motifs de recherche.



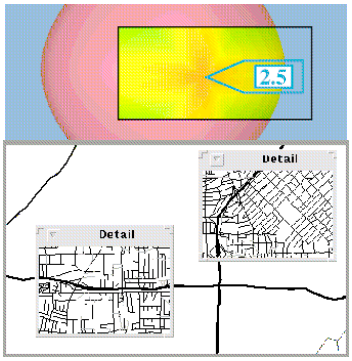
Loupe [Goscinny, Uderzo et al. 1961]



Fish-eye

```
    1 The FISHEYE view: a new look at structured files
    2      I. ABSTRACT
    3      II. INTRODUCTION
...23      III. GENERAL FORMULATION
...51      IV. A FISHEYE DEFINED FOR TREE STRUCTURES
   52         A. The Underlying Fisheye Construction and its Properties
...76         B. Examples of Fisheye for Tree Structured Files
   77            1. Indent Structured Files: Structured Programs, Outlines, etc.
   78               a. Examples: Programs, Outlines, etc.
   79               b. Usually ordered - fisheye is compatible
   80               c. Specific example 1: paper outline
>>81                  Figure 3: outline, regular and fish views
   82                  i. some adjacent info missing
   83                  ii. traded for global information
   84               d. Comment: standard window view = degenerate fisheye
   85               e. Specific example 2: C program code
...89               f. Other indent structures: biol. taxon., org. hierarch...
   90            2. Count-Until: A Simple Generalization of Indent Structure
...100           3. Examples of the Tree Fisheye: Other Hierarchical Structures
...106      V. FISHEYE VIEWS FOR OTHER TYPES OF STRUCTURES
...117      VI. A FEW COMMENTS ON ALGORITHMS
...140      VII. OTHER ISSUES
...162      VIII. CONCLUDING REMARKS AND SUMMARY
```

Fish-eye logique [Furnas 1981]



Lentille magique [Bier, Stone et al. 1993]

Figure 6.18 –Distorsions optiques et logiques

Pour présenter ces lentilles métier, prenons un exemple : nous voulons obtenir un aperçu rapide des regroupements automatiques réalisés dans notre analyse de données d'expression de gènes. Pour cela, nous activons une lentille spécifique : pour un groupe donné, elle met en évidence les gènes qui lui sont associés, et elle affiche les annotations de ces gènes. Très

rapidement, on voit apparaître des groupes de termes corrélés qui font émerger la « *thématique* » du groupe (figure 6.19). Dans un système à base de chemins, on aurait produit une requête équivalente du type : *« groupe(x) → gènes → annotations »* qui aurait produit un résultat non visuel, ne faisant pas émerger de corrélation aussi simplement. De plus, il faudrait spécifier une nouvelle requête à chaque fois que l'on s'intéresse à un nouveau groupe.

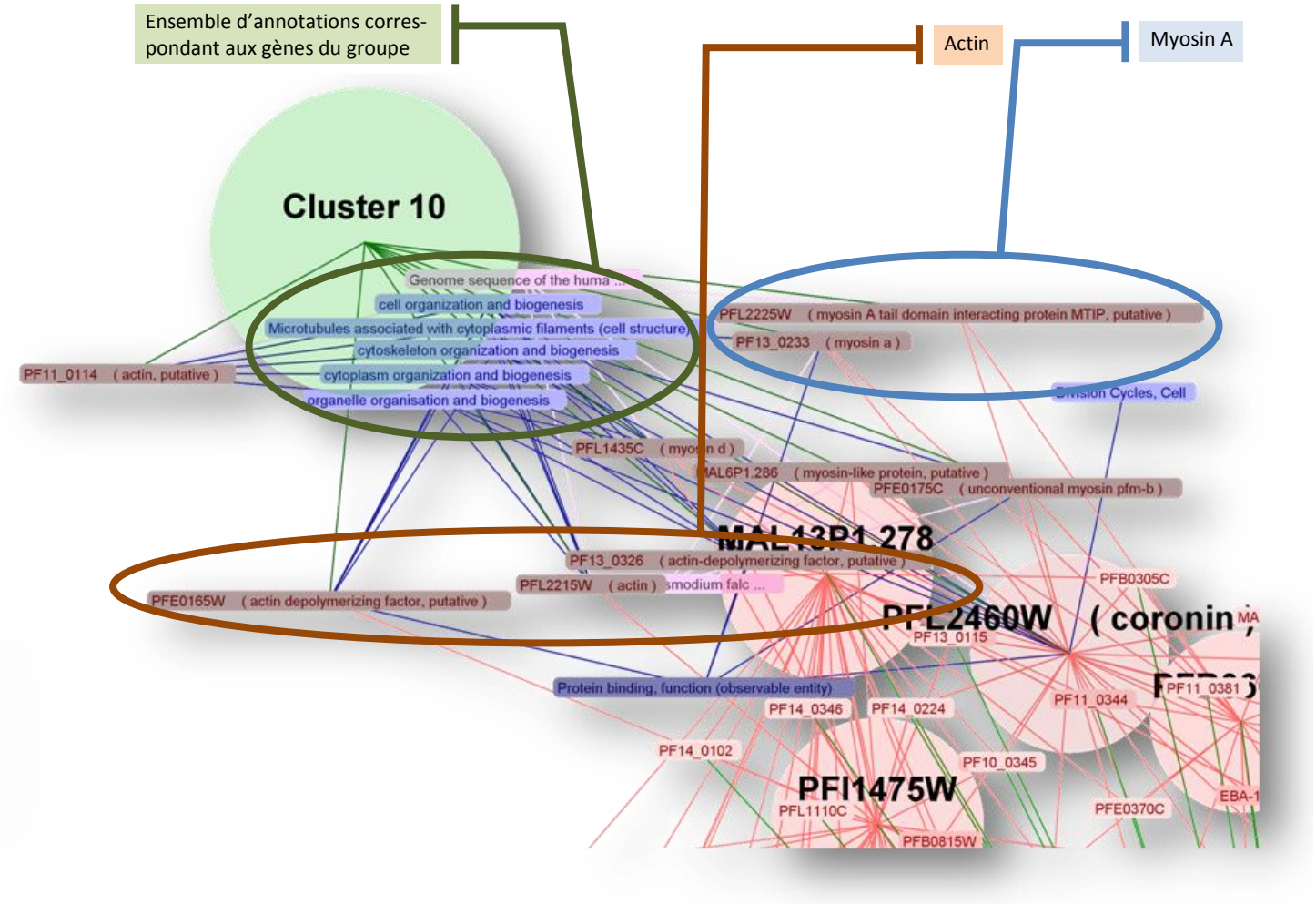


Figure 6.19 – Capture de l'application d'analyse de données d'expression : dans cette capture, une lentille est activée ; elle dispose autour d'un groupe de gène choisi l'information fonctionnelle relative (résumés, annotations). Les gènes sont en rose (lie de vin pour les gènes foncé par la lentille et appartenant au « Cluster 10 » sélectionné). Les concepts sont en bleu. Les clusters sont les cercles (rose pour les regroupements flous, verts pour les classements réalisés par Bozdech). Plusieurs observations peuvent être réalisées à partir de cette figure montrant l'émergence d'une information pertinente.
On voit apparaître deux protéines, l'actine et la myosine bien séparées. Ces deux protéines sont caractéristiques de fibres musculaires. Les annotations sont aussi cohérentes, et on distingue rapidement que ce « cluster 10 » regroupe des gènes qui caractérisent l'organisation structurelle de la cellule.

Afin de proposer une fonctionnalité générique et capable de s'adapter à des tâches métiers spécifiques, nous proposons d'assembler une lentille à partir de plusieurs composants génériques et atomiques. La spécification d'une lentille se fait à l'aide de plusieurs types d'objets :

- la lentille décrit le fonctionnement général,
- la sélection est la structure de données contenant les éléments sous l'effet de la lentille,
- le filtre permet de paramétrer la sélection et d'exprimer un motif de façon complète,

- des *modifieurs* sont des actions associées à la lentille qui lui confèrent son effet visuel.

Il existe plusieurs types de filtres génériques : certains filtres sont fréquemment utilisés (en fonction du type ou d'une classe par exemple), d'autres sont composites, permettent la combinaison de plusieurs filtres existants (composites ou non) pour exprimer des formules plus complexes : *et*, *ou* et *non* logiques.

De la même façon, les *modifieurs* permettent de spécifier le comportement de la lentille : certains permettent de foncer, d'éclaircir et de rendre opaques certains éléments. D'autres sont plus spécifiques : un *modifieur* permet d'afficher le descriptif résumé d'un gène en plus du nom. Un *modifieur* composite permet d'associer plusieurs effets à un élément de données.

Les scripts qui suivent permettent d'illustrer notre propos. Le premier permet de gérer simplement la sélection de plusieurs nœuds en cliquant et maintenant la touche « contrôle » enfoncée ou en réalisant un glisser-déposer pour délimiter une région rectangulaire (un exemple de codage est présenté dans la figure 6.20). Le deuxième est un outil de recherche (figure 6.21) : les éléments qui satisfont l'expression régulière spécifiée sont foncés, ceux éventuellement masqués sont rendus temporairement visibles. Enfin, la dernière est spécifique à l'analyse de données d'expression (figure 6.23) : lorsque l'on sélectionne un regroupement, tous les gènes qui lui sont associés sont mis en évidence, leurs annotations et les publications qui leur sont relatives sont alors disposées à proximité.

```
   // Gestion d'une sélection multiples et du retour utilisateur
 1 DefaultMultiNodeSelection selection=new DefaultMultiNodeSelection();
 2 this.main_selection= selection;

 3 SelectionRectangleDrawer gmsr= new SelectionRectangleDrawer(this);
 4 actionSubset.add(gmsr);
 5 MultiSelectionControl msc=new MultiSelectionControl(selection,gmsr);
 6 controls.add(msc);

 7 GenericLense multiselectlense = new GenericLense();
 8 actionSubset.add(multiselectlense);
 9 multiselectlense.coreNodeModifier=new DarkerModifier();
10 multiselectlense.selection= selection;
```

Figure 6.20 – Gestion de la sélection multiple

Ce premier exemple (figure 6.20) permet de gérer une sélection de nœuds de l'utilisateur comme cela existe dans les logiciels courants. L'objet `DefaultMultiNodeSelection` est la structure de données permettant de contenir les nœuds. Elle est affectée comme sélection principale de l'application (on peut envisager par la suite de gérer plusieurs sélections en parallèle). Nous proposons deux façons de sélectionner plusieurs nœuds : on clique successivement sur plusieurs éléments en maintenant une touche définie enfoncée, ou on définit une région rectangulaire sur la carte en réalisant un glisser-déposer. Dans les lignes 3 à 6, l'objet `MultiSelectionControl` gère cette interaction ; il est paramétré par la structure de données de sélection, et par un objet `SelectionRectangleDrawer`. Ce dernier est une action consistant en un retour utilisateur : durant le glisser-déposer, un rectangle transparent permet de visualiser l'étendue de la sélection. Le contrôle d'interaction et l'action en retour utilisateur sont ajoutés dans leur listes respectives. Enfin, la lentille est définie dans les quatre dernières lignes. Notons la spécification d'un modifieur `DarkerModifier` qui a pour effet de foncer les éléments sélectionnés.

```
    // déclaration du modifieur qui rend visible et assombrit les nœuds répondant à la
    requête
1   AggregateModifier visibleAndDarker=new AggregateModifier();
2   visibleAndDarker.addModifier(new VisibleModifier());
3   visibleAndDarker.addModifier(new DarkerModifier());

    // déclaraton d'une sélection (la structure de données)
4   this.searchSelection =new SearchSelection(this.graph);

    // déclaration de la lentille et ajout à la liste des actions
    // la lentille est en charge d'appliquer les modifieurs aux éléments de la sélection
5   this.searchLense= new SearchLense(this.searchSelection,visibleAndDarker,null);
6   actionSubset.add(searchLense);

    […]

    // lorsque l'utilisateur ouvre la boite de dialogue, la sélection est donnée en
    // paramètre à cette boite de dialogue (actuellement, la sélection est considérée
    // comme une propriété de l'application).
7   searchBar = new SearchToolBar(this.application);
```

Figure 6.21 – Script de spécification d'une lentille de recherche textuelle

Dans le second exemple (figure 6.21), on définit une lentille de recherche : une boîte de dialogue préalablement construite permet de saisir une expression textuelle (expression régulière). Dans la pratique, si on rentre une chaîne de caractères dans cette boîte de dialogue, les éléments dont le nom contient cette chaîne sont mis en évidence. Dans notre exemple, l'action réalisée par cette lentille est double : elle rend visible les éléments répondant à la requête s'ils sont masqués et fonce tous les éléments. On introduit donc un modifieur composite (`AggregateModifier`) dans lequel on insère une modifieur qui rend visible un élément et un second modifieur qui le fonce (resp. `VisibleModifier` et `DarkerModifier`). La quatrième ligne déclare une sélection de recherche. La ligne 5 instancie la lentille qui est finalement ajoutée à la liste des actions. Les interactions étant gérées par la boîte de dialogue contenant le champ de saisie de l'expression régulière, aucun gestionnaire d'évènement (contrôleur) n'est impliqué.

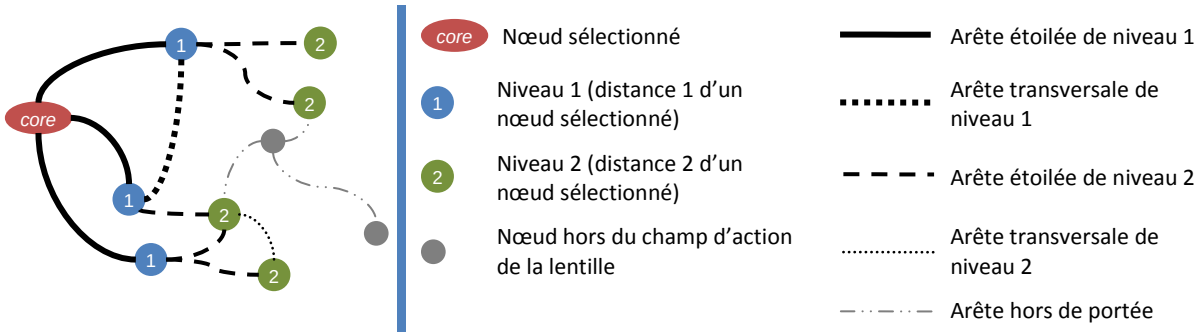


Figure 6.22 – Notion de niveau d'éloignement dans la lentille, arêtes directes et transversales.

La dernière lentille est plus complexe ; elle permet de répondre à des requêtes comparables aux chemins des systèmes à base de liens. Elle met en évidence pour un ou plusieurs regroupements choisis interactivement tous les motifs du type : *« groupe → gènes → (annotations | documents) »*[1] et de montrer les liens entre tous ces éléments. Par exemple, si deux sous-ensembles de gènes sont annotés différemment et si une arête « is-a » relie les concepts de ces annotations, alors la corrélation entre les gènes apparait à l'écran : les gènes sont reliés et regroupés autour d'un même concept. Cette corrélation serait moins visible sous forme d'une liste textuelle. Le motif est donc spécifié de la façon suivante (figure 6.22) : les nœuds sélectionnés font partie du « `core` », les nœuds à distance 1 et 2 de ce « `core` » sont dits de niveau 1 et 2. Les arêtes menant directement du « `core` » à un de ces nœuds sont dites étoilées de niveau 1 ou 2. Enfin, d'autres arêtes sont dites transversales car elles relient des nœuds à

[1] Le symbole « | » représente le « *ou* »

distance 1 ou 2 entre eux, mais ne correspondent pas à des chemins de longueur 1 ou 2 (respectivement) menant à ces nœuds à partir du `core`.

Ces termes, illustrés dans la figure 6.22, permettent de comprendre la nomenclature des objets du script suivant (figure 6.23). Les deux premières lignes déclarent la structure de données. Ici, la sélection est plus complexe : elle inclut automatiquement, à partir de nœuds centraux (`core`), les éléments à une distance 2 (`TwoLevelSelection`). Elle peut contenir des filtres (`Filtered`), et en l'état ne peut posséder qu'un seul nœud central (`SingleNodeSelection`). Il existe une version permettant de sélectionner plusieurs nœuds. La seconde ligne spécifie que l'on prend en charge les arêtes transversales. Si cette option était fausse, on pourrait ainsi restreindre le motif, non plus à un graphe mais à un DAG.

La seconde partie du script définit les filtres correspondant au motif de sélection de la lentille. Le premier filtre (`MultipleClassFilter`) n'autorise que certaines classes. Ici, il est dédié aux nœuds centraux et ne permet à la lentille de s'appliquer que sur des `NodeCluster` (regroupements) ou des gènes (qui sont aussi des regroupements flous). Le second filtre (ligne 6) permet de limiter l'extension du motif au premier niveau avec des arêtes correspondant à des regroupements. Le troisième filtre (ligne 7) permet de restreindre les nœuds de premier niveau uniquement à des gènes. Les lignes 8 à 10 restreignent les nœuds à distance 2 des regroupements à des documents ou des concepts. Enfin, les arêtes transversales ont un filtre les limitant à des annotations et des relations sémantiques. Ces filtres ont été instanciés, mais pas encore mis en relation avec la sélection. Ceci est réalisé de la ligne 14 à la ligne 19.

Dès lors, la sélection et son filtrage ont été dûment déclarés. Il faut instancier la lentille, associer des interactions et des effets. La lentille générique à deux niveau de profondeur est nommée `GenericLense`. Elle est initialement désactivée, l'utilisateur peut l'activer dans un menu. Comme dans les exemples précédents, elle est ajoutée dans la liste des actions. Nous introduisons ensuite de nouveaux modifieurs composites. Le premier est destiné aux arêtes ; il les rend visibles si elles ne le sont pas et rend leur force active si ce n'est pas le cas. Le second est destiné aux gènes : il fonce leur couleur et modifie leur présentation : au lieu d'afficher uniquement le nom du gène, il affiche leur résumé, ce qui correspond au *titre* ou au *produit* dans la figure 3.1 (page 82) : suivant le même exemple, on remplace *« PF11_0344 »* par *« PF11_0344 – Apical membrane antigen 1 precursor, AMA1 ».*

```
   // Création d'une sélection à deux niveaux de profondeur
 1 DefaultFilteredTwoLevelSingleNodeSelection selection=
                                    new DefaultFilteredTwoLevelSingleNodeSelection();
 2 selection.setComputeTransversalEdges(true);

   // Déclaration des filtres
 3 MultipleClassFilter corenodefilter=new MultipleClassFilter();
 4    corenodefilter.addClass(KMPNodeGene.class);
 5    corenodefilter.addClass(KMPNodeCluster.class);

 6 SimpleClassFilter l1edgefilter=new SimpleClassFilter(KMPEdgeClustering.class);

 7 SimpleClassFilter l1nodefilter=new SimpleClassFilter(KMPNodeGene.class);

 8 MultipleClassFilter l2nodefilter=new MultipleClassFilter();
 9    l2nodefilter.addClass(KMPNodeConcept.class);
10    l2nodefilter.addClass(KMPNodeDocument.class);

11 MultipleClassFilter l2transversalfilter= new MultipleClassFilter();
12    l2transversalfilter.addClass(KMPEdgeSemantic.class);
13    l2transversalfilter.addClass(KMPEdgeBoolean.class);

   // Paramétrage de la sélection par les filtres
14 selection.setCoreNodeFilter(corenodefilter);
15 selection.setLevelOneStarEdgeFilter(l1edgefilter);
16 selection.setLevelOneNodeFilter(l1nodefilter);
17 selection.setLevelTwoNodeFilter(l2nodefilter);
18 selection.setLevelOneTransversalEdgeFilter(l2transversalfilter);
19 selection.setLevelTwoTransversalEdgeFilter(l2transversalfilter);

   // Instanciation de la lentille
20 GenericLense lense = new GenericLense();
21 mouseoverlense.setEnabled(false);
22 actionSubset.add(lense);
23 lense.selection=selection;

   // Modifieurs
24 AggregateModifier visibleAndProcessing=new AggregateModifier();
25 visibleAndProcessing.addModifier(new VisibleModifier());
26 visibleAndProcessing.addModifier(new ProcessingModifier());

27 AggregateModifier genemodifier=new AggregateModifier();
28 genemodifier.addModifier(new DarkerModifier());
29 genemodifier.addModifier(new GeneNameModifier());

   // Paramétrage des effets de la lentille
30 lense.coreNodeModifier=genemodifier;
31 lense.firstLevelNodeModifier=genemodifier;
32 lense.firstLevelStarEdgeModifier=new DarkerModifier();
33 lense.firstLevelTransversalEdgeModifier=visibleAndProcessing;
34 lense.secondLevelNodeModifier=new VisibleModifier();
35 lense.secondLevelStarEdgeModifier=visibleAndProcessing;
36 lense.secondLevelTransversalEdgeModifier=visibleAndProcessing;

   // Gestion de l'interaction
37 SingleClickSelectionControl lensecontrol=new SingleClickSelectionControl(selection);
38 controls.add(lensecontrol);
```

Figure 6.23 – Script de spécification de lentille d'analyse fonctionnelle de gène

Les lignes 30 à 36 affectent à la lentille ces modifieurs :

- les gènes (regroupements flous ou associés à des regroupements) sont foncés et leurs étiquettes sont détaillées,
- les arêtes sont foncées,
- les documents et concepts sont rendus visibles
- les arêtes reliant documents et concepts sont rendues visibles et actives, ce qui a pour effet de disposer autour des regroupements les annotations et les documents.

Enfin, la dernière ligne permet d'affecter l'évènement du clic de souris à la lentille. 37 lignes ont donc été nécessaires pour déclarer cette lentille. De nombreux paramétrages sont possibles, et il est très simple d'étendre les interfaces des filtres, modifieurs, etc. pour créer ses propres outils. Il est possible de modifier l'action, le filtrage, le type de sélection ou l'évènement lié à une

lentille au cours de l'exécution et donc d'envisager que l'utilisateur définisse sa propre lentille interactivement, comme cela est proposé par exemple dans GenoLink [Durand, Labarre et al. 2006].

6.3.2.4 Deux nouvelles visualisations

La visualisation à base de ressort nous permet de générer de nombreuses cartes interactives. Grâce à une architecture souple, le développeur conçoit des cartes et des « clients riches » rapidement (en quelques heures) avec des fonctionnalités avancées. Les fonctionnalités précédentes visent à montrer l'intérêt de l'approche et la facilité avec laquelle le programmeur peut ajouter des fonctionnalités à son application. Par la suite, il est nécessaire d'enrichir ces fonctionnalités pour répondre aux besoins qui ne sont pas encore satisfaits. En particulier, nous avons relevé le besoin d'autres visualisations. D'une part, l'utilisateur a besoin de pouvoir se raccrocher à des visualisations plus courantes et consensuelles dans la communauté, d'autre part, la visualisation de regroupement que nous proposons fait disparaître la dimension temporelle des données. L'utilisation d'indicateur proposée par Michel Crampes ne serait alors pas adaptée [Crampes, Villerd et al. 2006]. Les indicateurs sont des vecteurs dimensions positionnables dans l'espace. Ils sont essentiellement utilisables en faible nombre simultanément, et chaque vecteur doit avoir une signification propre.

Afin de respecter les pratiques communautaires, nous introduisons une deuxième visualisation : les diagrammes d'Eisen (figure 1.18 page 28) [Eisen, Spellman et al. 1998], à partir de la bibliothèque existante MultiExperiment Viewer (figure 6.24). Cette visualisation, bien que fréquemment utilisée, est souvent critiquée (un constat réalisé à la suite de multiples entretiens avec des biologistes). Pour une analyse temporelle, les coordonnées parallèles sont plus adaptées pour faire apparaître des motifs et manipuler des données [Inselberg 1985]. Nous les avons intégrées en nous basant sur la librairie Parvis (figure 6.25) [Hauser, Ledermann et al. 2002]. Dans les deux cas, il est possible de synchroniser les interfaces en manipulant des sélections : la sélection d'un cluster dans la carte permet de colorier les vecteurs associés dans l'autre visualisation et de faire apparaître un motif grâce au principe de *brushing*.

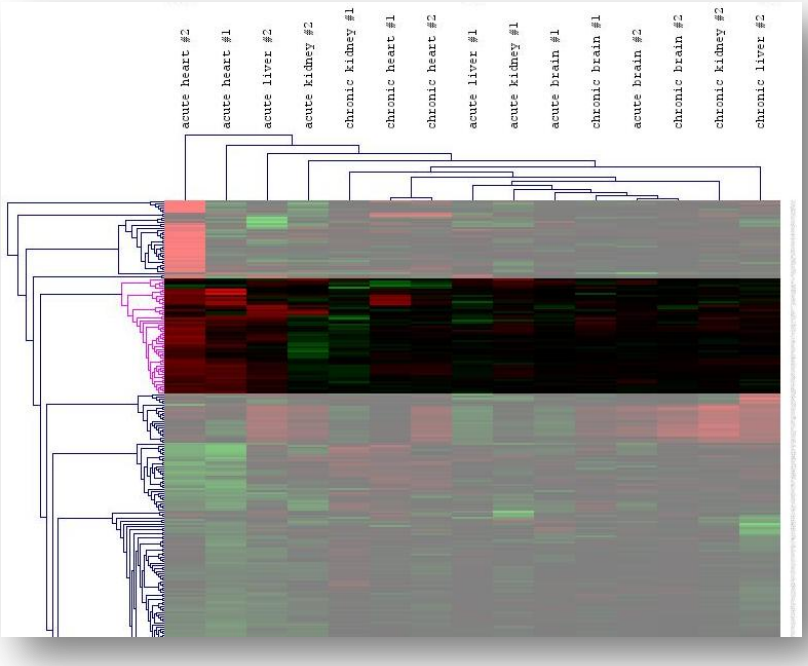


Figure 6.24 – Diagrammes d'Eisen repris de MultiExperiment Viewer : on peut sélectionner un ensemble de gènes, griser le reste, réordonner colonnes et lignes, afficher des dendrogrammes, cliquer sur chaque élément pour avoir une information détaillée, etc.

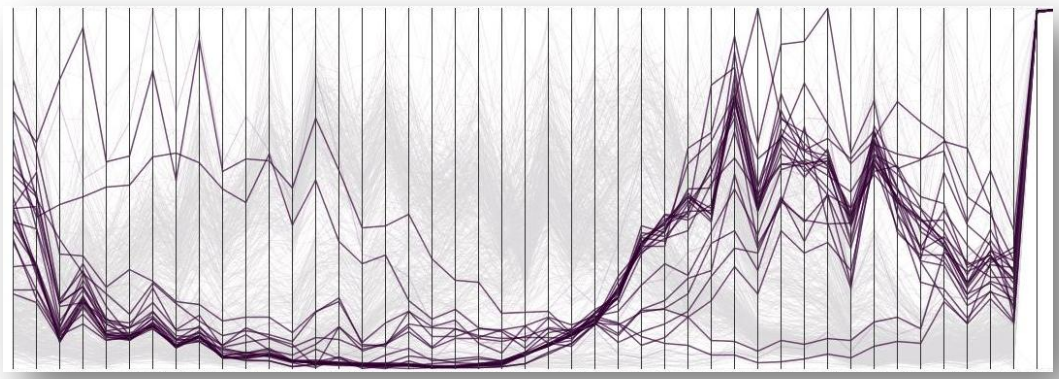


Figure 6.25 – Coordonnées parallèles intégrées à partir de Parvis : en fond, tous les vecteurs sont dessinés. Des régions denses apparaissent ainsi. Il est possible de colorier certains ensembles de gènes correspondant à un regroupement (plus généralement une sélection) afin de voir émerger un motif ou des anomalies sur un sous-ensemble de gènes.

6.3.2.5 Feuilles de style

Grâce aux fonctionnalités présentées jusqu'ici, l'utilisateur dispose de plusieurs applications interrogeant des données homogènes dans leur structure, leur contenu, et leur technique de visualisation. Cependant, différents développeurs peuvent proposer des applications très différentes. Si les couleurs des gènes et documents, par exemple, varient d'une application à une autre, l'utilisateur est perturbé. De même, les informations disponibles concernant un gène, un concept, etc. sont particulièrement nombreuses ; elles sont le plus souvent dépendantes de besoins spécifiques. Pour éviter à l'utilisateur la surcharge d'information telle qu'elle est présente dans les portails généralistes, il faut être capable de personnaliser au niveau d'un utilisateur et d'une application l'information souhaitée.

Pour cela, nous proposons la mise en œuvre de feuilles de styles. Ces feuilles permettent de centraliser et d'homogénéiser la description des caractéristiques d'une vue pour l'utilisateur. Elle directement liée aux métadonnées. On peut y décrire les couleurs des éléments, leur forme, leur rendu à l'écran, etc. Ces caractéristiques sont appliquées automatiquement à l'initialisation de l'application. Cependant, pour laisser toute liberté au développeur, il lui est possible de la modifier à volonté.

La feuille de style permet aussi de décrire le contenu souhaité : il est possible de sélectionner un ensemble de types d'attributs pour chaque type de nœuds et d'arêtes. Un panneau d'information affiche pour un élément sélectionné les attributs sélectionnés et leurs valeurs dans l'ordre de leur énumération. Un éditeur graphique permettra à l'utilisateur de configurer ce fichier grâce à la disponibilité des métadonnées.

A termes, cette feuille de style se révèle pratique dans d'autres contextes : dans le cadre d'un service en ligne de type portail, l'utilisateur peut l'utiliser pour spécifier quels attributs doivent être affichés et dans quel ordre. Dans le contexte de services Web, cela permet de spécifier et restreindre les données à transférer. Enfin, nous prévoyons son utilisation pour définir une structure de tableau permettant l'export d'un ensemble de données vers un tableur en générant un fichier automatiquement.

6.4 Adaptabilité

Les données manipulées dans I^{2}DEE sont présentes en grande quantité et leur usage est varié. Il est nécessaire de pouvoir adapter les données de l'entrepôt au contexte de l'utilisation. L'adaptabilité est alors présente à tous les niveaux de l'environnement. Dans la section précédente, nous avons montré comment la visualisation était capable de s'adapter à des tâches

spécifiques en permettant de construire des applications spécialisées et en permettant d'adapter la vue en cours d'utilisation. Les résultats visuels obtenus sont présentés et discutés dans le chapitre suivant. Dans cette section, nous présentons les mécanismes d'adaptation des données qui sont présents à divers niveaux de l'environnement : au sein de l'entrepôt de données, au sein de l'application cliente, ou encore au sein du service d'extraction de cartes contextuelles.

En premier lieu, nous présentons le mécanisme d'extraction de carte qui s'inspire des algorithmes d'analyse de liens utilisés par les moteurs de recherche récents (PageRank, Hits, etc.) [Borodin, Roberts et al. 2005]. Nous proposons ensuite de compléter ce mécanisme par un critère d'utilisation des données et discutons l'élargissement de ces critères. Nous concluons enfin sur les intérêts et les limites de cette approche, en détaillant comment la spécification de cette adaptation peut être simplifiée pour le développeur dans I^{2}DEE, en particulier en étendant la notion de feuille de style.

6.4.1 Extraction d'une sous-carte

Présentation générale

Les données de l'entrepôt sont volumineuses : dans le cadre de notre expérience sur Plasmodium Falciparum, l'entrepôt contient plusieurs millions de nœuds et quelques dizaines de millions d'arêtes. Dans le contexte de données génomiques sur des organismes réunissant des communautés plus larges, l'ordre de grandeur serait nettement supérieur : GenBank contient plus de 60 millions de séquences différentes et PubMed référence près de 44 millions d'auteurs différents pour 14 millions de documents. La mise en œuvre d'un entrepôt est trop lourde pour l'utilisateur final ; il est nécessaire de favoriser la mutualisation des données pour des raisons de coûts mais aussi pour limiter l'hétérogénéité des données. L'implantation d'entrepôts de référence permet de pallier les phénomènes de divergences des métadonnées et des identifiants. Nous dissocions donc l'entrepôt de cartographie des connaissances biologiques de la carte qui propose une vue sur une partie des données.

La carte dépend du domaine d'étude, de l'organisme concerné et de la tâche à effectuer. Pour être contextualisée, on doit disposer d'informations décrivant le contexte, dont la disponibilité varie en fonction du client. Ces informations sont peu prévisibles. Par exemple, un système de recherche d'information n'a connaissance que de quelques mots-clés correspondant aux requêtes. Dans un gestionnaire bibliographique, on dispose d'une liste de documents, et de la fréquence à laquelle l'utilisateur y accède ou les cite. Un outil d'analyse de données d'expression de gènes ne possède initialement qu'une liste de sondes présentes sur la puce à ADN (ou tout autre support) et connait l'organisme de référence. Nous avons donc choisi de laisser la liberté au développeur d'utiliser des méthodes d'extraction de cartes contextuelles existantes ou d'en introduire de nouvelles. Une telle méthode se définit simplement par le fait qu'elle retourne un sous-ensemble des tuples de l'entrepôt. La méthode que nous proposons dans la suite a été développée directement en Java. A termes, il serait préférable d'introduire un réel service Web permettant une complète indépendance et accordant plus de souplesse.

Analyse de liens

> *A good authority is one that is pointed to by many good hubs, and a good hub is one that points to many good authorities. [Kleinberg 1998]*

« *L'analyse de liens* » dont nous parlons ici ne fait pas référence aux systèmes à base de liens décrits par S.C. Boulakia, mais aux techniques mises en œuvre récemment par les principaux moteurs de recherche, dont Google [Kleinberg 1998; Page, Brin et al. 1998] (cf. [Borodin, Roberts et al. 2005] pour un état de l'art complet). Ces techniques sont fondées sur le constat que les documents répondant à une requête doivent avoir un contenu pertinent mais doivent aussi « faire autorité ». La première hypothèse établie considère qu'un fort degré entrant dénote une popularité de la cible [Marchiori 1997]. Kleinberg montre la distinction entre *popularité* et *autorité* [Kleinberg 1998]. Brin & Page observent alors que les liens n'ont pas tous le même

poids. L'autorité d'une page est d'autant plus grande que les pages la référençant font aussi autorité. D'un point de vue algorithmique, ils mettent en œuvre un mécanisme de propagation du poids d'une autorité à ses successeurs directs. Cette méthode est appelée PageRank et a donné naissance à Google. Kleinberg modère cette hypothèse et affirme que la présence d'un lien direct entre deux autorités n'est pas indispensable. Il étend la propagation de poids aux voisins de distance 2 dans le graphe. Cela aboutira au système Hits (Hypertext Induced Topic Selection). Une page assume alors simultanément deux identités :

- le « hub » capture l'utilité d'une page par le fait qu'elle référence des pages utiles et faisant autorité,
- l'autorité d'une page capture l'utilité propre d'une ressource par le fait qu'elle est référencée par des ressources de qualités (de bons *hubs*).

Plusieurs mesures sont proposées pour produire une note à partir de ces deux identités de la page. De multiples évolutions existent, prenant par exemple en compte le contenu textuel.

Justification du choix de cette méthode

Dans l'entrepôt, les nœuds ne sont pas tous des documents, ce sont des données atomiques. Les nœuds pertinents, initialement, sont ceux correspondant directement aux descriptions du contexte : ce sont des mots-clés (concepts), des documents, des gènes, etc. L'utilisation d'un périmètre de distance élémentaire pour étendre le graphe explose rapidement en combinatoire. Par exemple, par l'intermédiaire des types sémantiques et des mesures de cooccurrence, un concept possède un voisinage à distance 2 de l'ordre de plusieurs centaines de milliers d'éléments. Dans le cadre du projet Plasmodium Falciparum, par exemple, 500 000 nœuds sont présents à une distance 2 des 500 gènes présents sur la puce à ADN. Dans certains cas, il est possible de prendre en compte un critère de pertinence. Par exemple, il est possible d'analyser le contenu des documents ou d'exploiter une distance sémantique entre des concepts [Ranwez, Ranwez et al. 2006]. Ces critères de pertinence sont cependant spécifiques à certains contenus et certaines applications.

Dans le contexte de notre entrepôt de graphe, on conserve des hypothèses voisines : un élément est d'autant plus utile qu'il est lié à d'autres éléments, et que ces éléments sont utiles. Plusieurs différences existent cependant :

- certaines arêtes ne sont pas orientées, il est donc difficile de parler de degré entrant ou sortant,
- les nœuds et arêtes du graphe sont typés, une information qui nous permet de proposer des hypothèses sur la topologie et l'utilité de certaines données,
- les arêtes sont valuées, il est possible d'utiliser des hypothèses sur la source, sur la preuve ayant permis d'obtenir la relation, sur la valeur de cooccurrence, etc.

Les méthodes d'analyse de liens offrent ainsi de multiples avantages. Elles sont intuitives pour le développeur qui souhaite paramétrer ou créer une nouvelle méthode. Ceci est d'autant plus important que les données du contexte n'ont parfois jamais été expérimentées pour une telle application dans la littérature. Par exemple, nous n'avons pas connaissance d'un outil d'ingénierie ontologique ou d'un outil de gestion bibliographique qui se base sur une liste de gènes présents sur une puce à ADN. De plus, ces méthodes sont simples et rapides à mettre en œuvre. Elles permettent d'ordonner les éléments et de choisir la taille de la carte de façon continue. Robustes, elles sont applicables dans des topologies diverses de graphes et dans le contexte où l'évaluation, de la pertinence par une analyse du contenu n'est pas possible.

Exemple d'application

Pour faire la preuve de l'intérêt de cette méthode, nous avons fait le choix de l'appliquer dans un contexte extrême : l'information que nous possédons sur le contexte de l'utilisateur est une liste de gènes présents sur une puce à ADN. Nous souhaitons dès lors extraire une carte à des fins diverses : analyser les données de cette puces et construire une ontologie relative à ce

contexte. Les données utilisées proviennent des expériences de Bozdech [Bozdech, Llinás et al. 2003] sur *Plasmodium Falciparum*.

A partir des 500 gènes listés dans le fichier contenant les données d'expression, nous incorporons tous les nœuds situés à une distance maximale de 2 dans le graphe. Pour cela, nous ne prenons pas en compte les relations de cooccurrence, ni les distances *ultramétriques*[1] responsables d'une très forte connexité. Malgré ces précautions, le graphe résultant est composé de 500 000 nœuds : les types sémantiques d'UMLS représentent des hubs, certains sont reliés à plusieurs dizaines de milliers de nœuds.

Pour réduire le nombre d'éléments, nous appliquons un algorithme de pondération des nœuds basé sur la propagation du poids d'un nœud à ses voisins : les gènes centraux sont initialisés avec un poids de 1, les nœuds à une distance de 1 de ces gènes avec un poids de 0,5, les autres avec un poids nul. En procédant itérativement à partir du centre du graphe, chaque sommet propage sa pondération (*rank*) à ses voisins. Soit n_i un sommet et n_j un de ses voisins :

$$rank(n_j) \leftarrow rank(n_j) + \frac{rank(n_i)}{degree(n_i)}$$

Nous avons choisi de conserver les 5 000 nœuds possédant le plus fort poids. Cette valeur peut sembler arbitraire, nous la justifions de la façon suivante : cette valeur est 10 fois plus grande que le nombre de gènes initialement présents. Or, les gènes possèdent généralement de l'ordre de 3 annotations et de 2 à 5 liens bibliographiques. Certaines annotations et certains documents sont communs à plusieurs gènes. Nous conservons donc un voisinage important au regard de la taille des données utiles. Après un nombre d'itérations assez faible, nous avons ainsi la garantie de conserver les voisins directs des gènes. Après avoir déterminé ce sous-ensemble de nœuds, nous complétons la carte :

- en ajoutant les nœuds parents à partir des principales relations hiérarchiques (*« est un »*, *« partie de »*, etc.),
- en ajoutant toutes les relations à forte connexité ignorées jusque là.

La procédure complète est schématisée dans la figure 6.26. La sous-carte que nous utilisons dans la suite possède près de 6000 nœuds, dont approximativement 2000 sont des concepts, 2000 des gènes et 1000 des documents. Les résultats visuels sont présentés dans le chapitre qui suit. Nous n'avons pas mené d'évaluation formelle ou quantitative de la méthode, il est en effet difficile de proposer une carte *« de référence »* le permettant. Du point de vue de performances, l'exécution de cette procédure ne prend que quelques dizaines de secondes. La programmation de cette procédure a été réalisée directement en Java au niveau du serveur. L'extraction d'une carte à un instant donné n'exclut par son extension future ou l'accès parallèle à des données de l'entrepôt par l'intermédiaire de l'API.

[1] Distance calculée par la longueur du plus court chemin entre deux concepts dans Gene Ontology.

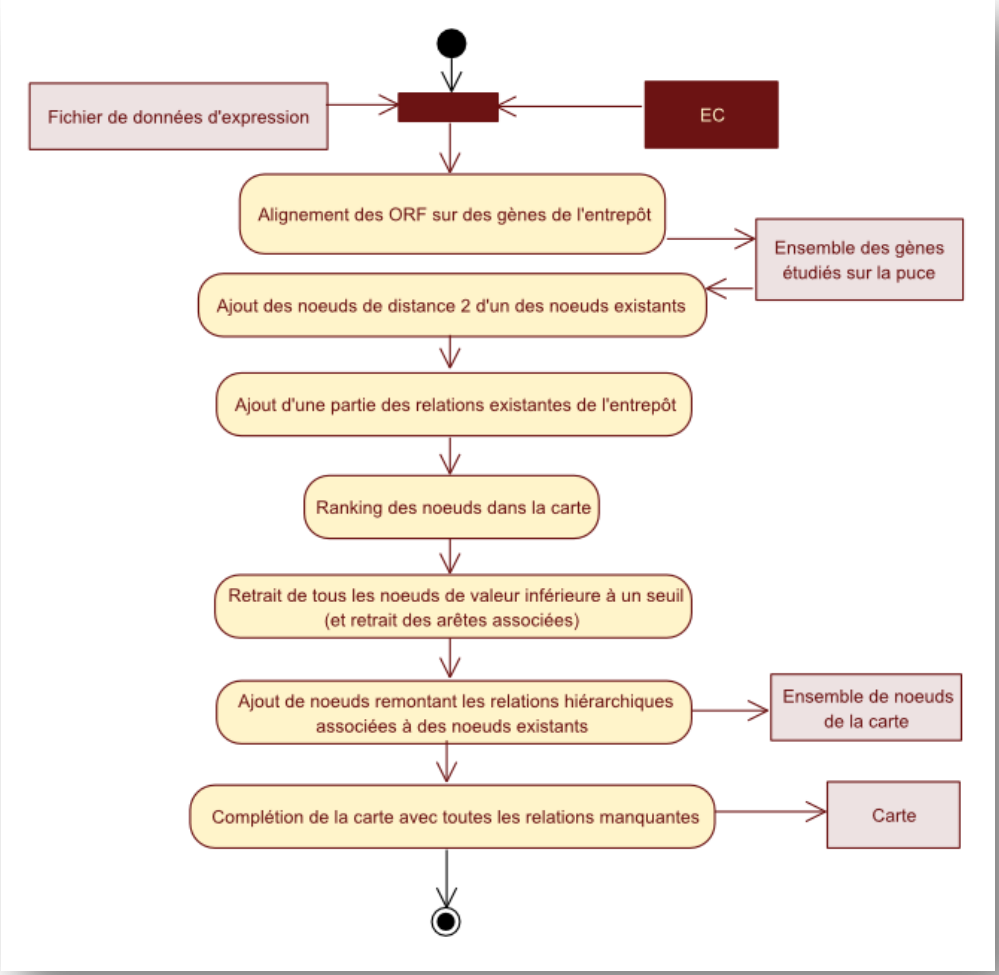


Figure 6.26 – Procédure d'extraction d'une carte contextuelle.

6.4.2 Adaptabilité de la carte et gestion des préférences

6.4.2.1 S'adapter à l'usage des données

La méthode d'extraction permet d'obtenir une portion contextuelle de l'entrepôt. Afin de s'assurer d'une information complète, nous faisons le choix de conserver plus d'éléments que nécessaire. Cette étape vise essentiellement à fournir à l'utilisateur un ensemble de données complet et capable d'être contenu dans la mémoire d'une station de travail courante.

Face à une surcharge d'information pour l'utilisateur, il est nécessaire d'adapter l'information visualisée à son besoin. Les critères structuraux ne suffisent pas : dans notre cas, le degré entrant d'un nœud peut être l'indice d'une utilité de ce nœud. Mais cela n'est pas toujours le cas : par exemple, dans le cadre de *Plasmodium Falciparum*, les publications les plus référencées correspondent aux projets de séquençage. Concernant une analyse fonctionnelle des gènes, elle n'est pourtant pas pertinente. L'évaluation de la pertinence du contenu n'est pas toujours possible, et les informations concernant le contexte sont pauvres. La solution que nous proposons est donc d'adapter les données par une apprentissage du système basé sur l'utilisation qui en est faite.

Le principe est comparable à celui présenté dans la section précédente : certaines actions ajoutent ou retranche du poids à certains nœuds, une propagation peut être réalisée. Pour présenter l'utilité de ce mécanisme, prenons un exemple concret : dans un outil d'analyse de

données d'expression, Pierre s'intéresse à un regroupement, consulte les informations sur certains gènes et leur ajoute des commentaires. Au travers des menus contextuels, il choisit d'ouvrir des pages vers un portail du domaine. En effectuant ces actions, le regroupement sur lequel est appliquée une lentille de façon prolongée voit son poids augmenter. Les gènes pour lesquels Pierre ajoute des commentaires voient aussi leur poids croître fortement. Ces augmentations sont propagées sur le voisinage : les gènes appartenant au regroupement, les concepts correspondant aux annotations et les documents relatifs à ces gènes sont aussi mis en valeur. D'autres gènes ne sont pas associés au regroupement sur lequel se focalise Pierre ; ceux associés à des annotations similaires seront tout de même mis en évidence, au-delà du processus d'analyse de données, grâce au principe de propagation.

Supposons maintenant que Jean, qui appartient à la même équipe que Pierre, soit en charge de construire une terminologie du domaine. Il utilise une application d'ingénierie terminologique et ontologique. Les concepts qui ont été au centre de l'attention de Pierre sont mis en valeur. Jean clique sur un concept, et choisit un terme alternatif ou une définition : on peut en déduire que le terme est pertinent, car, dans le cas contraire, il l'aurait supprimé. Les concepts qui sont disposés autour de ce concept ont probablement été lus par Jean. S'il ne les a pas supprimés, ils sont potentiellement pertinents. La propagation peut donc se faire par un voisinage dans le graphe ou par un voisinage à l'écran. La propagation peut donc être conditionnée par la visibilité à l'écran et la proximité sur le plan projeté.

Ces pondérations ne concernent pas seulement certains nœuds et s'appliquent à des métadonnées. Lorsque Jean structure sa terminologie, il utilise la relation « est un » de façon exclusive, ou choisit de conserver d'autres relations présentes. L'interface peut à terme en déduire l'intérêt de Jean pour certains types de relation. La propagation peut dépendre ou non d'une relation. Si un concept est considéré comme pertinent et si on lui attribue un poids fort : le poids des hyperonymes est fortement augmenté, le poids est hyponymes l'est aussi dans une moindre mesure. Si Jean a éprouvé jusqu'ici un intérêt pour la relation de méronymie, on peut aussi appliquer la propagation des poids au travers de cette relation. Si Jean supprime un concept, on peut pénaliser dans une faible mesure les hyperonymes, et on peut proposer de supprimer directement tous les hyponymes du concept ou plus généralement les concepts qui seraient déconnectés du graphe. La sélectivité de la propagation, dans le contexte de l'analyse de données d'expression, peut s'illustrer de la façon suivante : lorsque Pierre porte un intérêt particulier pour un concept, on propage cet intérêt plus fortement vers les concepts annotés manuellement par ce concept que vers les gènes annotés automatiquement.

6.4.2.2 Autres pondérations

Nous avons présenté deux pondérations : une première est structurelle (ou topologique) et une seconde est liée à l'utilisation des données et aux interactions de l'utilisateur. D'autres critères peuvent être utilisés pour pondérer l'intérêt de certains éléments dans carte. Nous les regroupons dans trois catégories :

- les critères distributionnels,
- les critères de pertinence,
- les critères de fiabilités et de qualité reconnus.

Les critères distributionnels sont obtenus à partir des procédures de fouille décrites dans le chapitre 5 (section 5.5 page 150). Par exemple, TF.IDF[1] est une pondération proposée par Salton dont l'intérêt a été montré par le passé [Salton and Buckley 1987; Manning and Schütze 1999].

L'évaluation de la pertinence consiste à représenter une adéquation sémantique entre le contenu et le contexte. On peut par exemple pour cela utiliser des représentations vectorielles [Rocchio 1971; Lafourcade, Prince et al. 2002] complémentaires des approches à base de liens en recherche d'information [Bianchini, Gori et al. 2005].

[1] Term Frequency × Inversed Document Frequency

Enfin, les derniers critères sont des critères relatifs à la réputation d'une partie des données dans la communauté. Certaines ressources sont considérées comme plus fiables ou plus précises du point de vue des données, annotations, etc. Ce critère permet de décrire qu'une donnée provenant de RefSeq est plus fiable qu'une donnée uniquement présente dans GenBank, ou encore qu'une annotation issue de PlasmoDB serait plus fiable qu'une annotation générée automatiquement par une méthode d'alignement.

A quoi s'appliquent ces critères ?

Les critères structuraux s'appliquent essentiellement aux nœuds. Ils sont obtenus par la propagation de poids entre les nœuds et au travers des liens. Nous n'avons pas perçu d'intérêt à la propagation de poids entre des métadonnées (type, source, etc.). En revanche, il est intéressant d'agréger les valeurs des ces critères en fonction de certaines métadonnées, ou de limiter la propagation en fonction des ces données (sources, preuve, type, etc.). Par exemple, dans le contexte d'une étude de réseaux sociaux des auteurs de PubMed, on peut souhaiter limiter cette pondération à des auteurs, et la propagation à des relations de co-écriture d'article (chemins acycliques de longueur 2 dans le sous-graphe composé des auteurs et des documents). Cela permet par exemple d'utiliser des mesures de centralité courantes dans ce domaine. Dans le cadre de l'ingénierie ontologique, on peut souhaiter consulter au niveau d'une ressource complète le cumul ou la moyenne des valeurs afin de comparer l'utilité de différentes ressources entre elles.

Les critères distributionnels s'appliquent aux nœuds. Une valeur correspond à l'occurrence répétée d'un terme dans un ou plusieurs documents. A nouveau, il peut être souhaitable pour diverses raisons d'agréger les valeurs à l'échelle d'une ressource : quelles sont les sources les plus fréquemment utilisées ? Dans quelle mesure sont-elles généralistes ou spécialistes ? Lesquelles apparaissent dans des contextes similaires ? La propagation entre des métadonnées ne nous parait pas utile.

Les critères de pertinence proposent une mesure de similarité entre des nœuds et un contexte. L'application de ce critère se fait à l'instar des deux précédents : elle s'effectue à l'échelle des données et non des métadonnées ; on peut, dans un but introspectif, agréger les valeurs relatives à certaines métadonnées.

Au contraire, les critères de fiabilité et de réputation s'appliquent essentiellement à des métadonnées. Un concept d'UMLS par exemple est plus fiable qu'une chaîne de caractères extraite de façon répétée dans le corpus. Les données de RefSeq ou de PlasmoDB sont plus fiables *a priori* que celle de GenBank ou Entrez Gene respectivement. Une annotation manuelle peut être considérée comme plus sûre qu'une annotation automatique. Enfin, on peut souhaiter s'intéresser aux connaissances sur plusieurs espèces : dans le contexte médical humain par exemple, il peut être souhaitable de privilégier l'information issue d'expérimentation sur des cellules humaines à celle provenant d'expériences réalisées sur des souris.

On peut discuter l'application de ce type de critères pour certains types de nœuds : la réputation d'un auteur est exploitée quotidiennement, celle d'une revue ou d'une conférence aussi (*« impact factor »*, etc.). Cependant, la cotation d'une revue est généralement réalisée par une institution de référence (ISIKnowledge par exemple), on peut représenter cette valeur comme un attribut du journal ou de la conférence. Le calcul de la cote d'un journal, d'un article ou d'un auteur est généralement basé sur leur référencement : un article est d'autant mieux coté qu'il est publié dans une revue cotée, et qu'il est référencé par des articles eux mêmes cotés. Un journal a d'autant plus d'intérêt qu'il contient des articles bien cotés (et par conséquent bien référencés). Un auteur est d'autant plus réputé qu'il publie des articles référencés fréquemment dans des journaux prestigieux. Ce principe de calcul rejoint en fait les méthodes d'analyse de liens regroupées dans les critères structurels.

Enfin, les critères d'utilisation des données s'appliquent aux deux niveaux indépendamment : nous avons déjà montré comment l'action porte directement sur des gènes, des concepts, des regroupements, et leur voisinage. Ces actions peuvent aussi avoir en parallèle des répercutions

sur des métadonnées : lorsque l'utilisateur s'intéresse à certains attributs plus qu'à d'autres, cela permet d'adapter la vue, de réordonner ces attributs, de masquer ceux qui paraissent inutiles, etc. De la même façon, si l'utilisateur ne s'intéresse pas aux publications, aux locus et aux protéines, mais uniquement aux gènes et à leurs annotations, il est possible d'adapter la vue. Lorsqu'il privilégie systématiquement une définition ou un terme provenant d'une même source, le logiciel peut prendre en compte cette préférence automatiquement, à l'échelle d'une ressource.

6.4.2.3 Implémentation de ces critères

Les données des critères précédents sont stockées dans des relations distinctes des données de l'entrepôt. Pour l'instant, chaque critère donne lieu à une relation. La nomenclature automatique et normalisée de la relation permet d'ajouter ou supprimer un critère sans le coder statiquement (*« en dur »*). Cependant, les opérations pour modifier les valeurs sont dépendantes du critère.

L'application du critère d'utilisation peut se faire à deux niveaux : au niveau du serveur, les procédures stockées métier permettent d'encapsuler ces opérations en toute transparence. On peut paramétrer l'API pour activer ou désactiver cette fonctionnalité. Lors de la construction initiale de l'entrepôt, ou lors de mises à jour, on désactive cette fonction. Lorsque l'application cliente souhaite stocker au niveau du serveur le critère d'utilisation (pour mutualiser l'usage au sein d'un laboratoire par exemple), le développeur l'active. Ces opérations peuvent aussi être directement encapsulées dans les interactions du client graphique :

- elles peuvent être spécifiques à une application,
- elles peuvent être affinées en fonction de l'interaction réalisée dans l'application cliente et pas simplement de l'opération d'accès ou de modification présente dans l'API,
- elle est applicable lorsque l'on ne dispose pas d'un serveur et donc pas de procédures stockées.

6.5 Synthèse

Nous venons de présenter la couche interactive de notre environnement. Elle répond aux principaux besoins que nous avons énumérés dans la section 6.2.1. En positionnant des points fixes, on peut dessiner les différentes parties d'une cellule, et leur rattacher des mots clés et les gènes correspondant. Le fond de carte, figé ou non, peut être différent du graphe dessiné par-dessus. La projection de données multidimensionnelles nous est apparue satisfaisante du point de vue du respect des distances et du voisinage. L'évolution de la vue est dynamique : l'utilisateur peut interagir en continu pour filtrer les données, annoter des éléments, etc. Des contrôleurs permettent d'adapter la vue et de la faire évoluer en fonction des besoins. Les feuilles de styles permettent de spécifier le contenu que l'on souhaite visualiser et d'homogénéiser les interfaces des utilisateurs, qu'il s'agisse de clients riches ou de portails.

Les données systèmes d'information du domaine peuvent être intégrées et visualisées en lien avec les données expérimentales du chercheur. Les lentilles métiers permettent de proposer des « requêtes interactives », une alternative visuelle aux systèmes à base de liens et aux langages de requêtes traditionnels.

La boîte à outils est extensible, sa mise en œuvre est simple et en lien avec les données de l'entrepôt. Pour fédérer le développeur, elle propose de nombreuses fonctionnalités pour accéder aux systèmes d'information externes, gérer des vues multiples, des sélections, exporter des données (captures d'écrans, export vers des tableurs, etc.). La plupart des graphes que nous visualisons ont un nombre d'arêtes linéaires en fonction du nombre de nœuds, ce qui permet de visualiser des graphes de l'ordre de plusieurs milliers de nœuds avec fluidité.

La quantité de données présentes dans l'entrepôt est nettement supérieure aux capacités de notre boîte à outils et ces données sont difficilement manipulables par l'utilisateur. Nous avons mis en œuvre un mécanisme d'extraction de carte permettant à partir d'un algorithme d'analyse de lien d'isoler une portion utile de l'entrepôt. Les données sont adaptées à une station de travail courante et à la boîte à outils graphique, mais il subsiste une surcharge cognitive pour l'utilisateur. Des mécanismes d'adaptabilité complémentaires sont proposés, et qui permettent de prendre en compte :

- les préférences de l'utilisateur au niveau des métadonnées,
- la distribution des données,
- les interactions de l'utilisateur,
- des critères de pertinence.

Tous ces critères d'adaptabilité s'intègrent correctement dans la méthode de visualisation qui permet une adaptation de la vue continue. L'application de ces mécanismes n'est pas sans rappeler le fonctionnement d'une lentille et une requête sous-forme de chemin. XSL(T) permet de transformer une structure de documents XML et pas uniquement de définir un « style ». De la même façon, nous prévoyons d'adapter ou d'étendre la syntaxe d'une feuille de style afin que le développeur puisse définir un patron pour l'extraction d'une carte, et pour la restriction de la propagation d'un critère.

Nous n'avons pas mené d'évaluation précise de ces critères d'adaptabilité, mais leur utilisation passée dans des moteurs de recherche comme Google, ou encore dans des indices d'intérêt (*Impact factor*) ont montré leur robustesse. Le chapitre suivant présente les résultats graphiques et des cas d'utilisations dans deux contextes applicatifs : l'analyse de données d'expression et l'ingénierie terminologique et ontologique. Ces résultats sont commentés et discutés.

CHAPITRE 7

Applications et résultats visuels

« Interfaces emerge when the system is used »

KARI KUUTTI

7.1 Introduction

Dans ce qui précède, nous avons présenté l'environnement I²DEE suivant différents niveaux : niveau général, conceptuel, architectural ou encore au niveau de l'implémentation. Nous avons décrit I²DEE comme une approche visuelle souple, extensible, capable de s'adapter dans des contextes variés pour réaliser diverses tâches. Pour témoigner de cette capacité d'adaptation et de la pertinence de notre approche, nous proposons dans ce chapitre d'explorer deux contextes applicatifs très différents correspondant à des besoins identifiés d'utilisateurs :

- l'ingénierie terminologique et ontologique [Jalabert, Ranwez et al. 2006],
- et l'analyse de données d'expression de gènes [Jalabert, Ranwez et al. 2006].

La première application cible un utilisateur bioinformaticien, terminologue ayant des connaissances en biologie et maitrisant la modélisation. Cet utilisateur peut être amené à construire des ontologies spécifiques. Cependant, de nombreuses ressources existent déjà et couvrent une partie de son problème. I²DEE doit lui permettre de construire une ontologie plus rapidement en réutilisant des ressources existantes et de gagner en qualité de résultat (confiance et cohérence) dans la mesure du possible.

La seconde application cible un utilisateur biologiste souhaitant analyser des données d'expression de gènes obtenues *via* un dispositif haut débit. Il souhaite procéder à un regroupement automatique des gènes puis en explorer les résultats afin d'extraire un petit ensemble de gènes d'intérêt. L'application lui permet de visualiser deux regroupements simultanément (cf. section 6.2.1) et doit permettre à l'utilisateur de consulter rapidement la connaissance du domaine de façon contextuelle. Ce chapitre présente ces deux applications, discute les résultats obtenus, et dresse un bilan de ces deux expériences.

7.2 I²DEE comme support à l'ingénierie terminologique et ontologique

Concevoir une terminologie ou une ontologie est une tâche longue est difficile. Différentes méthodologies et différents environnements existent pour supporter les experts dans cette tâche. Produire une RTO est un travail qui doit être incrémental et collaboratif pour que l'ontologie résultante fasse référence dans le communauté. Dans le contexte biomédical, le constat actuel est que les ressources terminologiques et ontologiques (RTO) sont de plus en plus nombreuses et leur taille croît. Dans la mesure du possible la conception doit être accomplie en réutilisant des ressources existantes. Cependant, la plupart des environnements ne sont pas adaptés : la réutilisation de ressources existantes se restreint à l'import depuis des formats standards et la navigation dans l'arbre est généralement restreinte à un dessin horizontal (cf. figure 7.1).

Dans la suite de cette section, nous abordons tout d'abord les principes généraux de la conception d'une terminologie et en détaillons les différentes étapes. Nous nous intéressons, ensuite, aux environnements intégrés dédiés aux dernières étapes de la conception : formalisation et évaluation. Nous dressons finalement un bilan de ces outils et montrons comment I²DEE peut s'avérer une approche pertinente, adaptée au contexte biomédical. Nous détaillons alors quelques résultats visuels obtenus avant de conclure sur cette application.

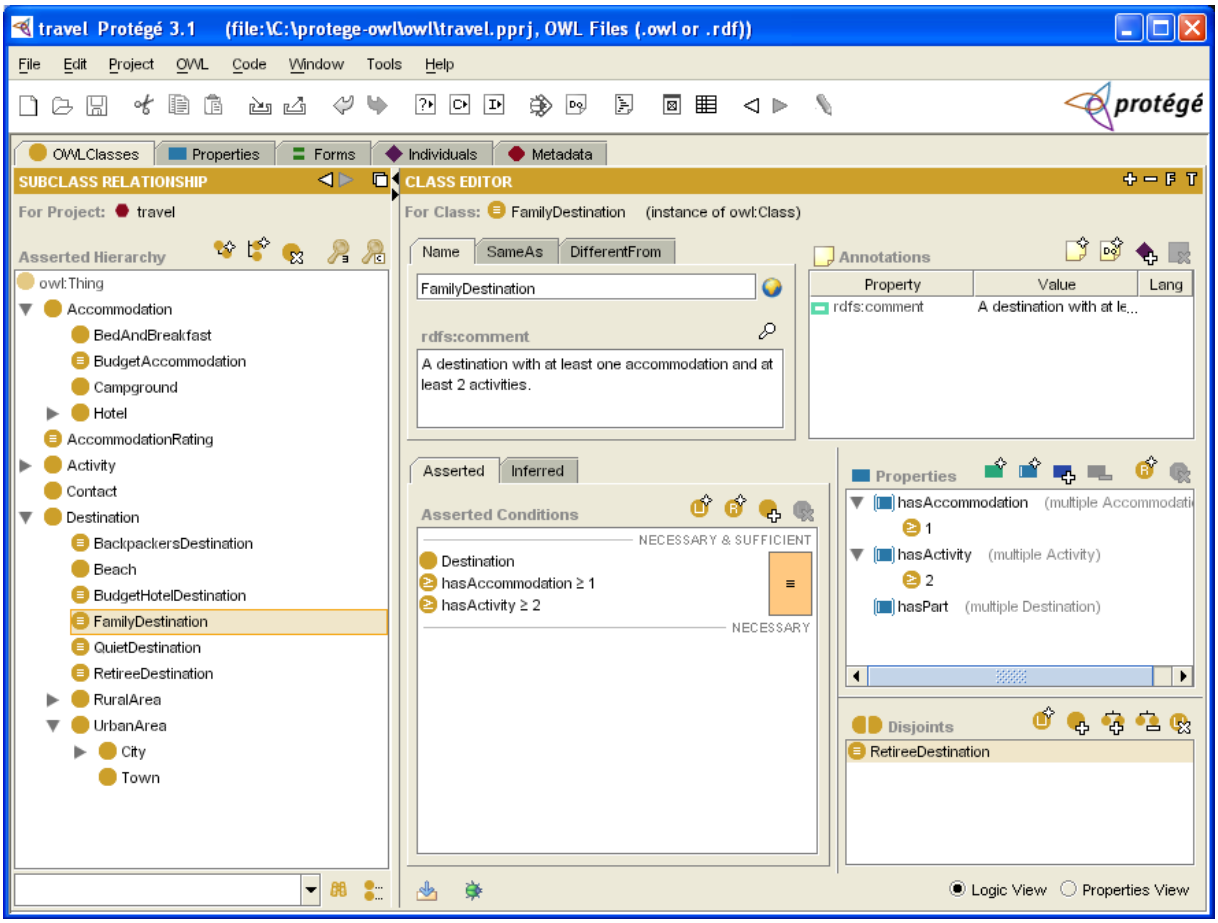


Figure 7.1 – Protégé OWL, éditeur de classes (concept), capture issue du site officiel de Protégé.

7.2.1 Principes généraux de conception de RTO

Il existe de multiples méthodologies pour concevoir des ontologies. Sans les détailler, deux grandes approches existent : la première particulièrement prisée au sein de la communauté française considère qu'une RTO ne peut être construite que manuellement en s'appuyant sur une expertise du domaine [Bachimont 2000; Bourigault, Aussenac-Gilles et al. 2003]. Elle s'avère, le plus souvent, difficilement réutilisable car elle est spécifique à une tâche et à une application. La communauté internationale, plus généralement, s'intéresse à la conception automatisée d'une RTO et ne la perçoit pas avec le même point de vue formel [Maedche 2002]. Au sein de la communauté des sciences du vivant, cette seconde approche domine : les ontologies sont constituées de milliers, ou de centaines de milliers de termes, et lorsque l'on observe le contenu d'UMLS, on constate la présence d'anomalies qui laissent supposer une utilisation importante d'automatismes.

Si les techniques mises en œuvre ne sont pas identiques dans ces deux approches, elles respectent globalement la séquence suivante :

- Analyse de corpus
- Construction de la RTO
  - Modélisation
  - Normalisation
  - Formalisation
- Mise en œuvre et retour d'expérience qui engendrent des corrections/évolutions

La première étape dans la conception d'une RTO est de constituer un corpus représentatif du domaine et adapté à l'utilisation qui en sera faite. Par exemple, dans le contexte de l'indexation documentaire, un échantillon du contenu de PubMed peut être utilisé. Concernant le problème

de la gestion de dossiers des patients en milieu hospitalier, certaines expériences précédentes ont fait le choix de quelques ouvrages et des rapports d'actes médicaux [Zweigenbaum 1999; Bourigault, Aussenac-Gilles et al. 2003].

Une fois ce corpus constitué, on lui applique des méthodes de fouille de données textuelles. La communauté française adopte souvent une démarche globalement commune consistant à effectuer une analyse syntaxique des données textuelles et à appliquer par la suite des analyses distributionnelles [Harris, Gottfried et al. 1989; Aussenac-Gilles and Bourigault 2003]. On lie alors souvent des termes partageant un « contexte syntaxique ». Par exemple, les candidats termes *« insuffisance rénale »* et *« détresse respiratoire »* sont associés car ils sont compléments des syntagmes suivants : *« prise en charge »*, *« apparition »*, *« installation »*, *« admettre en réanimation chirurgicale pour… »*, etc. De même, les verbes *« montrer »* et *« mettre en évidence »* sont associés car ils ont tous deux pour sujet : *« échographie »*, *« bilan infectieux »*, *« tomodensitométrie »*, *« artériographie »*, *« auscultation pulmonaire »*, etc.[1] [Aussenac-Gilles and Bourigault 2003; Bourigault, Aussenac-Gilles et al. 2003]. A partir de ces résultats produits sous forme de liste, l'expert constitue manuellement l'ontologie.

Dans le contexte international, on utilise plus généralement des analyses distributionnelles et des méthodes d'extraction de connaissances faiblement supervisées. On distingue ainsi dans un premier temps certaines méthodes amont qui extraient les termes et les entités nommées, des méthodes aval qui extraient des relations sémantiques, le plus souvent à partir de patrons ou de règles. Les travaux généraux portent notamment sur l'extraction de relations d'hyperonymie, mais dans le contexte biologique, de nombreuses relations spécifiques ont donné lieu à des études : interaction entre protéines ou entre gènes, régulation, effet des médicaments, etc. Ces dernières techniques ont par ailleurs été mises en œuvre suivant l'objectif de fouiller les données de PubMed et non dans le but initial de construire une RTO.

Qu'il s'agisse de l'approche française ou de l'approche automatisée de la communauté internationale, la supervision par un expert est présente, mais de façon plus ou moins soutenue. Une analyse de données textuelles précède cette supervision. Dans ce contexte, la communauté française dissocie trois étapes que sont la modélisation, la normalisation et la formalisation. La modélisation vise à structurer les candidats termes à l'aide des relations. La normalisation élimine certaines variations afin d'obtenir un résultat correspondant à un consensus des utilisateurs. Enfin, la formalisation intègre ce résultat dans un formalisme prédéfini, comme le standard actuel OWL (Ontology Web Language) qui dérive d'XML. Différents outils existent actuellement pour ces tâches. Les éditeurs les plus connus, comme Protégé ou OntoEdit ne concernent que la dernière étape visant à produire un fichier opérationnel. Les premières étapes nécessitent généralement des outils de traitement automatique de la langue comme Syntex et Upery [Bourigault and Fabre 2000; Bourigault 2002] et des interfaces utilisateur adaptées (Terminae [Biebow and Szulman 1999], DOE spécifique aux ontologies différentielles [Troncy and Isaac 2002], etc.)

En résumé, l'environnement de développement de RTO doit permettre la constitution d'un corpus, son analyse syntaxique, distributionnelle ou à base de règles, et faciliter la structuration du résultat par l'expert. L'évaluation après mise en œuvre de la ressource est difficilement automatisable, elle est généralement réalisée par l'expert, suivant une méthode spécifique au contexte applicatif, en dehors d'un environnement de conception d'ontologie. Cependant, certains environnements proposent des outils de vérification de la consistance de la RTO, lorsqu'elle dispose d'un niveau de formalisme élevé.

Le cycle de vie d'une RTO est généralement comparable à celui d'un logiciel. Sa conception est incrémentale et elle est amenée à évoluer : subir des corrections ou des extensions. Etant spécifique à un domaine et une application, elle est généralement décrite comme difficilement réutilisable. Cependant, dans le cas d'une évolution ou d'une réutilisation, on prend en compte les ressources préexistantes dès les étapes préliminaires de constitution du corpus.

[1] Exemples extraits de [Aussenac-Gilles and Bourigault 2003].

7.2.2 Environnements intégrés d'édition formelle et d'évaluation

Dans [Jalabert, Ranwez et al. 2005], nous dressons un état de l'art des environnements intégrés. Nous les appelons environnement d'édition formelle car nous faisons abstraction de tous les outils conçus pour piloter des méthodes d'analyse de données textuelles et d'extraction d'information. Nous nous intéressons ici aux éditeurs permettant de concevoir la ressource en

- filtrant les candidats termes pertinents,
- structurant la ressource,
- normalisant les termes,
- formalisant les propriétés,
- vérifiant la consistance du résultat.

Pour cela, nous les comparons suivant 4 axes fonctionnels : la possibilité de travail collaboratif, la méthodologie à laquelle se rapporte l'outil, la présence d'un moteur d'inférence ou de vérification de consistance, et enfin l'interface utilisateur. D'autres états de l'art, plus exhaustifs, montrent la multiplicité des outils existants [1] [Corcho, Fernández-López et al. 2003; Mizoguchi 2004].

Edition collaborative

Ontolingua [Farquhar, Fikes et al. 1995] et OntoSaurus [Swartout, Patil et al. 1996] reposent tous deux sur une architecture client/serveur permettant une conception collaborative au travers d'une interface HTML manquant quelque peu de convivialité. Ils se distinguent par leurs langages unificateurs, respectivement KIF (Knowledge Interchange Format) [Genesereth and E. 1992] et LOOM qui est une variante du premier. WebOnto [Domingue 1998] et Apollo [Koss 2002] possèdent une architecture identique mais proposent une interface plus conviviale en utilisant des Applets Java. WebODE [Arpírez, Corcho et al. 2001] enrichit les pages HTML d'Applets et propose une architecture trois tiers. KAON [Oberle, Volz et al. 2004] propose une infrastructure plus complète encore. Soulignons enfin l'approche originale d'Hozo [Kozaki, Kitamura et al. 2002] qui introduit la notion d'ontologie distribuée. Différents utilisateurs conçoivent différentes ontologies qui sont reliées, mettant en jeu des mécanismes de synchronisation : les modifications d'une ontologie sont proposées dans les autres, validées par leur propriétaire éventuel, l'interface utilisateur y est particulièrement conviviale.

Méthodologies

Certains environnements se positionnent par rapport à une méthodologie de conception. OilEd [Bechhofer, Horrocks et al. 2001] s'attache à présenter les aspects de OIL et permet l'utilisation de l'inférence avec FaCT [Horrocks and Sattler 2001; Tsarkov and Horrocks 2006]. OntoEdit s'oriente vers la méthodologie On-To-Knowledge dérivée de CommonKADS [Sure, Erdmann et al. 2002; Sure, Staab et al. 2004]. Hozo implémente la notion de concept-rôle dérivée des propositions de J.F. Sowa [Sowa 1995; Kozaki, Kitamura et al. 2002]. Enfin, WebODE repose sur la Methontology [Lopez, Gomez-Perez et al. 1999] et DUET sur le paradigme UML.

Architecture et fonctionnalités complémentaires

Des services d'inférence sont proposés dans de nombreux environnements (OilEd, OntoEdit, OntoSaurus, Protégé2000 [Gennari, Musen et al. 2003], WebODE, WebOnto, Hozo, Apollo) afin de vérifier la consistance de l'ontologie, parfois aussi pour la classification automatique des concepts (OldEd, OntoSaurus). Enfin, notons que différentes architectures permettent d'utiliser tantôt des services d'inférence, des API ou encore permettent le branchement de modules complémentaires : Protégé2000, WebODE et KAON.

[1] *http://www.xml.com/pub/a/2004/07/14/onto.html*
http://hcs.science.uva.nl/wondertools/index.html

Interface utilisateur

Excepté certains outils orientés conception collaborative basés sur des formulaires et pages HTML, les interfaces des outils cités sont généralement conviviales et reposent sur des formulaires et une visualisation arborescente de l'ontologie, comme Protégé par exemple (cf. figure 7.1). Remarquons l'éditeur de Hozo, KAON OIModeler et le module Mind2Onto s'ajoutant à OntoEdit qui se démarquent par leur effort de convivialité. Notons également les travaux autour des graphes conceptuels qui proposent des méthodes de spécification visuelles, avec par exemple les éditeurs CoGITaNT et CharGer. Enfin, certaines ontologies reposent sur le paradigme d'UML ; DUET est un éditeur dédié à ce formalisme.

7.2.3 Pertinence de l'approche I²DEE

Parmi tous les environnements cités, Protégé2000 est certainement le plus répandu : il rassemble une grande communauté d'utilisateurs et possède de nombreux modules et une documentation détaillée. Son architecture est non seulement extensible sous forme d'onglets et de *plug-in*, mais il permet aussi une utilisation locale, sous forme de client serveur, et fournit une API complète pour le développeur. D'autres éditeurs sont utilisés spécifiquement par certains projets. DAGEdit, par exemple, était l'éditeur utilisé par les concepteurs de GO à ses débuts ; c'est ainsi qu'il a obtenu sa notoriété dans le contexte des sciences du vivant et de la bioinformatique.

Quelle que soit la notoriété des éditeurs, plusieurs constats peuvent être dressés :

- ils ne prennent pas en compte la connaissance du domaine,
- ils ne sont pas adaptés à la réutilisation de plusieurs ressources où à des systèmes médiateurs comme UMLS,
- ils ne possèdent pas de visualisation multiéchelle adaptée à des ressources de grande taille,
- ils ont tous des restrictions importantes (adaptés à une structure d'arbre ou de DAG uniquement par exemple).

Nous pensons qu'I²DEE est adapté à des applications biologiques car il répond à ces différents besoins :

- il permet d'accéder aux connaissances du domaine, de prendre en compte les données expérimentales du chercheur, et intègre directement UMLS (et donc MeSH, Snomed, GO, etc.),
- l'interopérabilité avec un logiciel de gestion bibliographique permet de constituer et de mettre à jour le corpus facilement,
- il propose une interface utilisateur avancée, multiéchelle, capable de s'adapter aux différentes tâches que doit réaliser l'utilisateur et à de multiples topologies de graphe.

La section suivante illustre ces fonctionnalités d'I²DEE en présentant des résultats visuels à différentes échelles. Nous montrons comment l'interface permet d'adopter alternativement plusieurs points de vue correspondant aux différentes tâches qui composent la conception d'une RTO.

7.2.4 Résultats visuels

Cette présentation des résultats adopte une démarche expérimentale. Dans un premier temps, nous observons la structure du graphe à un niveau macroscopique. Ensuite, nous nous intéressons plus particulièrement à deux régions : une bulle formée par la relation *« partie de »* et la périphérie en forme grillagée structurée par la relation *« est un »*. Cela nous permet de vérifier la satisfaction d'un besoin d'adaptabilité dans la démarche d'exploration ou de conception d'une RTO.

Rappelons que la carte est actuellement extraite en appliquant l'algorithme présenté dans la section 6.4.1 (page 180) à partir d'une liste de gènes analysés sur une puce à ADN.

Point de vue macroscopique

La figure 7.2 montre à une échelle globale le graphe qui contient de l'ordre de 2000 concepts. Nous ne visualisons pas les gènes, ni les documents et nous ne prenons pas en compte les relations bibliographique. A cette échelle, il est impossible de lire l'ensemble des concepts, mais en affichant les relations sémantiques, on peut distinguer globalement trois grandes régions dans le graphe. La partie dense et centrale est principalement organisée autour des relations de cooccurrences. La structure des graphes de cooccurrence est fréquemment assimilée à celle d'un réseau social. Comme cela est décrit dans [Bennouas 2005], ce type de graphe bien que possédant un degré moyen faible est généralement difficile à dessiner. Les annotations recouvrent partiellement le centre, mais elles relient des concepts distincts. Enfin, les relations sémantiques associent un ensemble de termes plus structurés et mieux dessinés dans la périphérie. On distingue une forme grillagée sur le pourtour du graphe et une sorte de bulle (en bas à droite).

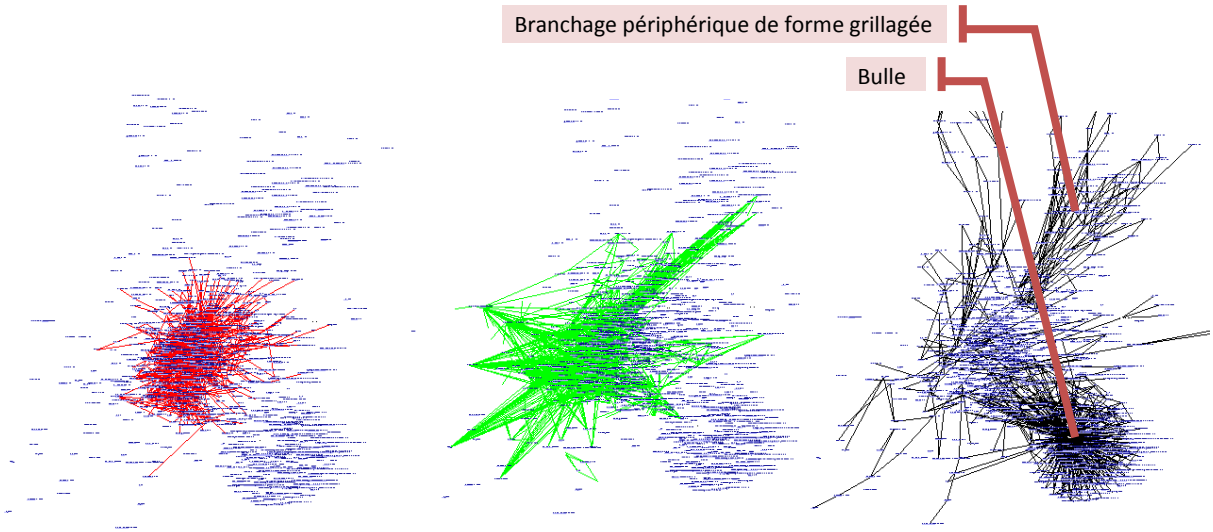


Figure 7.2 – Aperçu macroscopique de la carte et des trois principales relations : cooccurrences, annotations (entre un gène et un concept) et les relations sémantiques (« est un » et « partie de »).

La bulle « partie de »

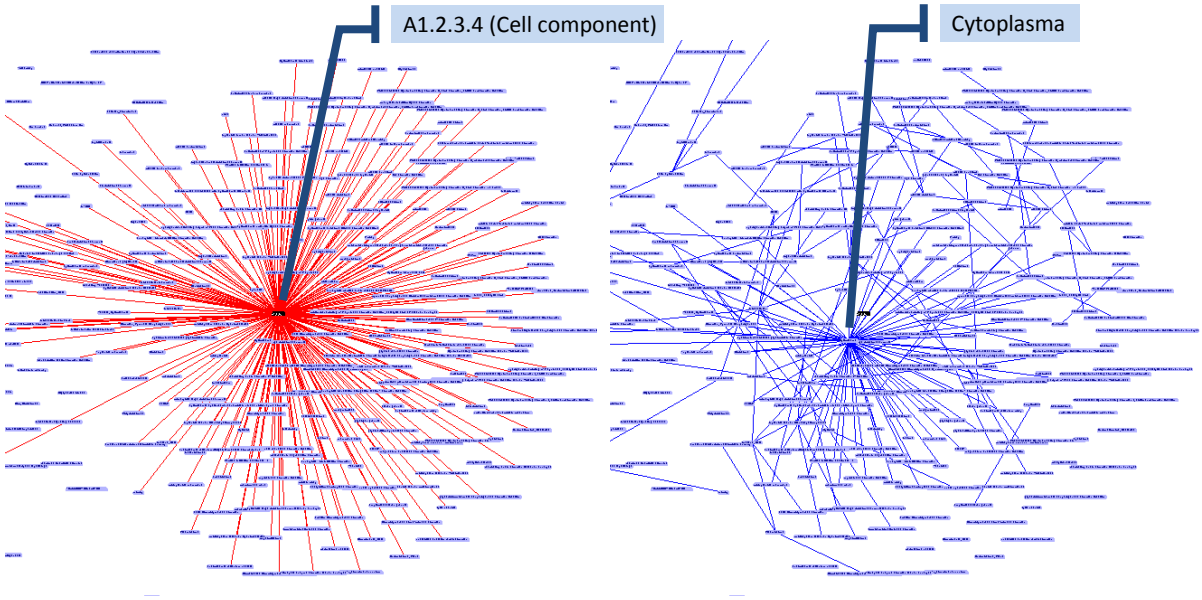
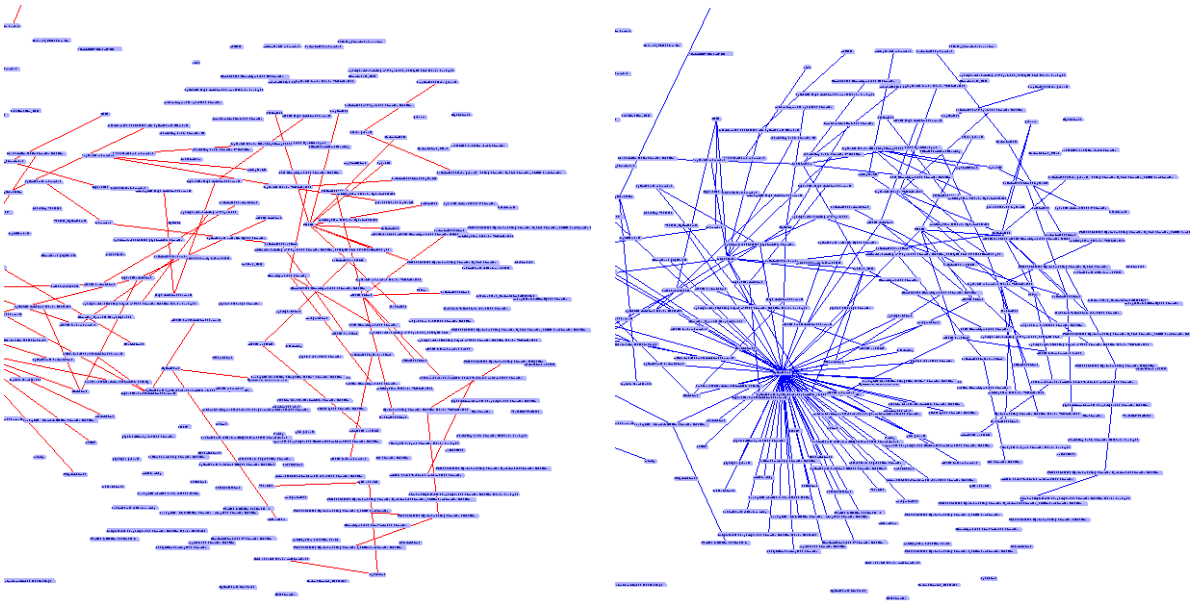


Figure 7.3 – Agrandissement du hub et des deux principales relations qui le structurent.

Dans la partie inférieure droite de la figure, on constate qu'une sorte de *bulle* se forme à partir des relations sémantiques. Nous nous y intéressons de plus près et avons choisi d'affiner la vue en dissociant les trois principales relations sémantiques : *« est-un »*, *« partie de »* et les *« type sémantiques »*. Dans la figure 7.3, seules les deux dernières sont présentées. On y observe que tous les éléments sont reliés à un type sémantique central. Lorsqu'on zoome, on lit que le concept central est *« cell component »* (identifié par A1.2.3.4 dans UMLS). La relation *« partie de »* n'a pas une forme étoilée aussi nette, mais un concept central proche du type sémantique présente un forte degré de connexité et produit aussi une étoile plus petite. Les autres relations sont moins denses et enchevêtrées.

La relation *« partie de »* et le type sémantique sont ici fortement corrélés. Cela s'explique par les deux centres *« cell component »* et *« cytoplasma ».* Ces deux éléments contiennent les différents composants de la cellule. La relation partie propose une représentation plus fine mais enchevêtrée.

Les figures que nous venons de détailler ont été produites en rendant visibles certaines relations successivement. La disposition des nœuds provient cependant des contraintes simultanément exercées par toutes les relations. Dans la figure 7.4, nous comparons la répartition des relations *« est un »* et *« partie de »*.

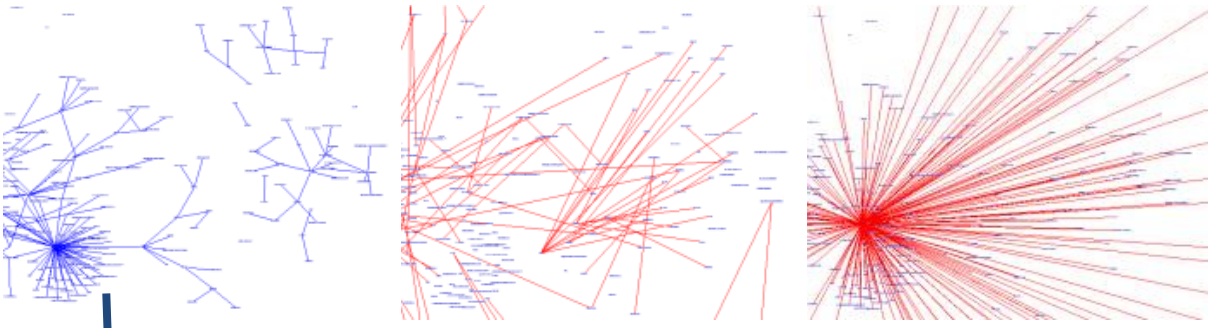


Relation « est un » *Relation « partie de »*

Figure 7.4 – Distinction de deux types de relation sémantique dans le hub.

Nous y observons que les enchevêtrements se situent dans les parties où les relations se recouvrent. La question que nous posons est de savoir s'il est pertinent d'exercer les forces selon plusieurs relations simultanément, ou s'il est préférable, dans certains cas, de limiter les forces actives.

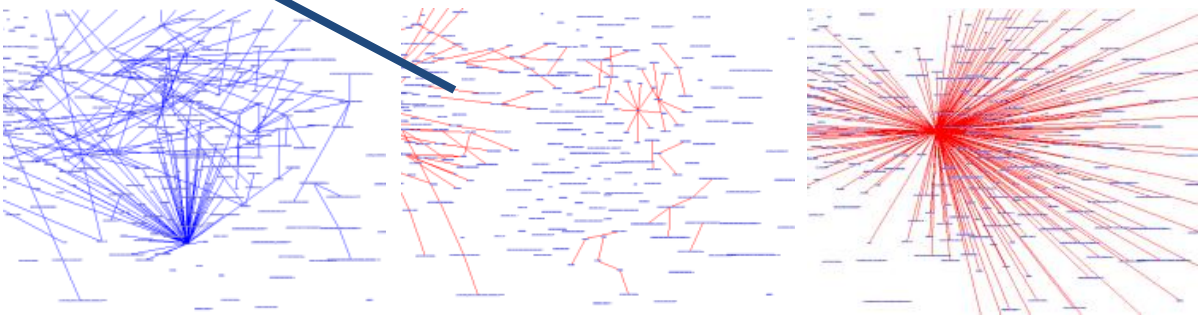
Dans la figure 7.5, nous disposons successivement les nœuds selon les contraintes exercées par une seule relation sémantique, et nous dessinons alors les différentes relations. La figure 7.5-a dispose ainsi le graphe selon la relation *« partie de »*. Le résultat du dessin nous permet de conclure : on obtient un dessin des relations presque planaire. On peut dissocier des composantes non connexes. Au contraire, les relations *« est un »* sont fortement enchevêtrées. Réciproquement, lorsqu'on dispose dans la figure 7.5-b les nœuds uniquement à partir de la relation *« est un »*, le dessin de cette relation est presque planaire, mais la relation *« partie de »* est fortement enchevêtrée. Dans les deux cas, le type sémantique conserve bien sur sa forme étoilée.



« partie de » « est un » Type sémantique

a – Disposition basée sur la relation « partie de »

Lorsque l'on dispose le graphe en appliquant les contraintes correspondant aux relations que l'on visualise, le graphe résultant est plus intelligible.



« partie de » « est un » Type sémantique

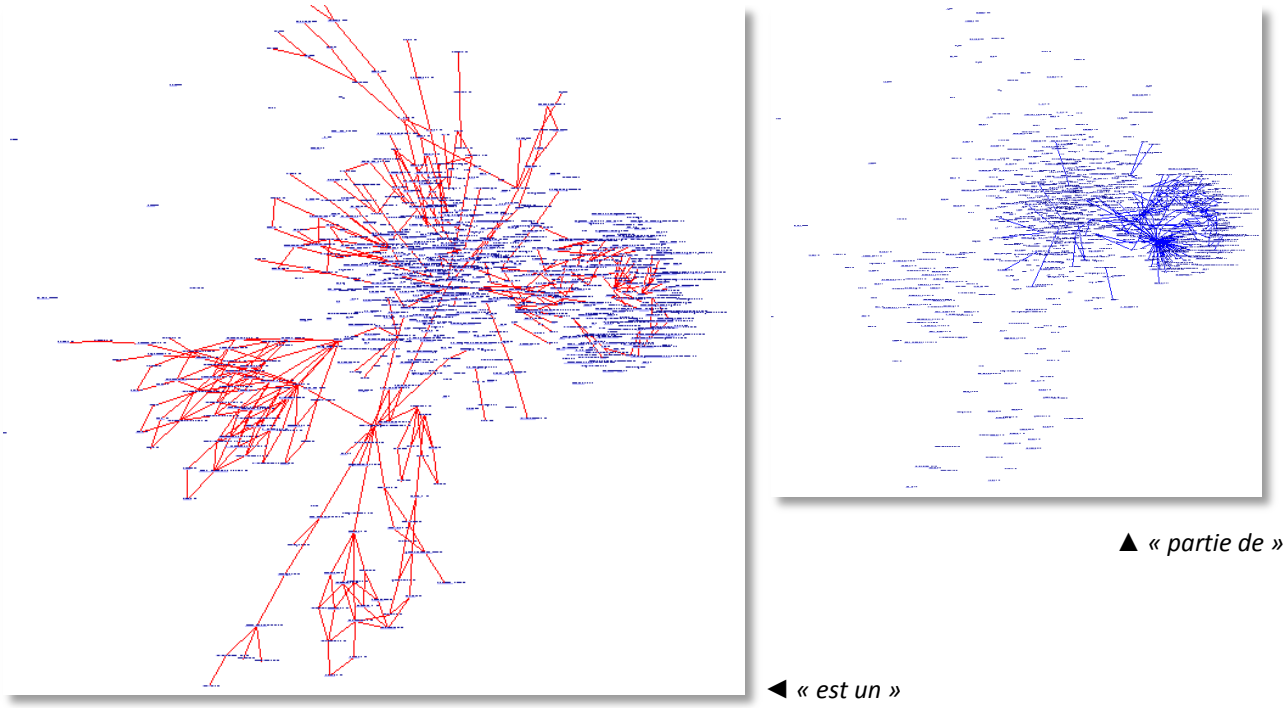
b – Disposition basée sur la relation « est un »

Figure 7.5 – Différentes vues de la même carte en fonction de types de relation différents.

Cette expérience assez simple nous montre l'intérêt de proposer plusieurs vues à l'utilisateur. Dessiner simultanément des relations différentes engendre un enchevêtrement. Au contraire, si on alterne les vues en exerçant les forces correspondant aux relations que l'on visualise, le graphe obtenu est nettement plus intelligible. Dès lors que l'utilisateur est amené à réaliser successivement différentes tâches dans la conception d'une RTO et à s'intéresser à différentes relations, il s'avère nécessaire de proposer une application capable de s'y adapter. C'est ce que fait I^2DEE.

La relation « est un »

La figure 7.6 montre la répartition des relations *« est un »* et *« partie de »* dans le graphe. La relation *« est un »* est responsable des branches et grillages dans la périphérie. On constate que les termes les plus fréquents et généraux sont au centre, les plus spécifiques étant plus présents à l'extérieur.



▲ « partie de »

◀ « est un »

Figure 7.6 – Répartition des relations "est un" et "partie de" dans le graphe.

Cette forme grillagée résulte dans notre expérience de la prédominance de Gene Ontology qui, rappelons le, n'a pas une structure d'arbre mais de « DAG ». Si ce n'est pas le cas dans cet exemple, cela peut aussi résulter de la présence de plusieurs ontologies contradictoires. La figure 7.7 propose un agrandissement correspondant à l'exemple que nous avons choisi dans la section 2.2.3.2 (page 46). La visualisation permet ainsi le localiser facilement les structures qui ne sont pas arborescentes dans ou les contradictions entre ontologie. L'utilisateur possède une vue qui lui permet rapidement de prendre les décisions adéquates en dissociant les options de généralisation choisies par les différentes sources.

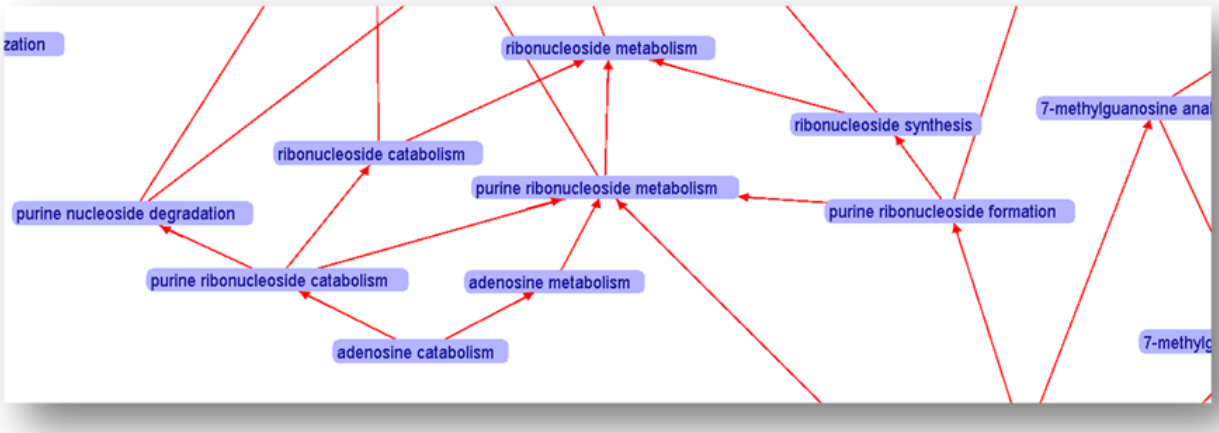


Figure 7.7 – Agrandissement d'une région périphérique grillagée structurée par « est un ». On retrouve les données correspondant à l'exemple présenté dans la section 2.2.3.2.

7.2.5 Bilan

La conception d'un RTO se compose de tâches multiples exprimant des besoins visuels différents. L'environnement de conception doit être capable d'explorer une (ou plusieurs) ressource(s) existante, de grande taille éventuellement. Il doit permettre de les dessiner quelle que soit leur topologie et d'en faciliter la réutilisation. De plus, le résultat produit doit souvent

contenir plusieurs relations sémantiques dont les plus courantes sont « est un » et « partie de ». L'environnement est capable de proposer alternativement une vue adaptée. Nous venons en effet de montrer que combiner plusieurs relations sémantiques ne permettait généralement pas d'obtenir une visualisation satisfaisante. Nous venons de voire que I²DEE proposent des fonctionnalités permettant de satisfaire techniquement ces exigences.

7.3 I²DEE utilisé en analyse de données d'expression

7.3.1 Le besoin

Contexte

Le besoin d'analyse de données d'expression a été exprimé à plusieurs reprises dans le cadre de nos collaborations. L'institut Pasteur a mené des expériences sur le génome de Plasmodium Falciparum et Y. Cayre sur l'homme. De plus, l'équipe rattachée au projet Arabidopsis du CEA a produit des données similaires dans un contexte protéomique : il ne s'agit pas à proprement parler de données d'expression, mais, de la même façon, on quantifie la présence de différentes protéines à différents instants ou dans différentes conditions expérimentales.

Positionnement

Jusqu'ici, la plupart des outils d'analyse de données effectuent un regroupement et produisent un résultat graphique ou non (diagrammes d'Eisen basés sur une classification hiérarchique, *plotting* dans un plan pour l'ACP, etc.). L'automatisation est alors restreinte à un calcul mathématique appliqué aux données. Le biologiste doit par la suite effectuer le travail le plus difficile : donner un sens biologique à ce résultat mathématique.

> *The challenge no longer lies in obtaining gene expression profiles, but rather in interpreting the results to gain insights in biological mechanisms. [Subramanian, Tamayo et al. 2005]*

Différentes démarches existent pour intégrer une information fonctionnelle dans ces méthodes. Certains travaux proposent de dessiner l'expression des gènes dans les graphes métaboliques, d'autres d'exploiter les résultats d'une fouille de données textuelles dans PubMed. Une autre solution consiste à utiliser les connaissances partagées par la communauté, et plus précisément les annotations fonctionnelles dans GO. Nous avons recensé deux méthodes récurrentes. La première consiste à déterminer un ensemble de concepts de GO qui sont anormalement présents (du point de vue des probabilités) dans le jeu de données et à étiqueter un jeu de données global par un ou plusieurs concepts probables. La seconde démarche consiste à normaliser une distance[1] dans le DAG de GO et à la fusionner avec la matrice de distances des profils d'expression.

Ces deux méthodes sont fortement automatisées, mais de notre point de vue produisent un résultat trop opaque pour l'utilisateur. Il est important d'intégrer l'information fonctionnelle et le profile d'expression qui doivent guider et améliorer l'analyse de données d'expression. Mais l'utilisateur doit conserver une vue directe sur ses données d'expression. La carte que nous proposons répond à ce problème en intégrant visuellement les informations essentielles associées aux données expérimentales de l'utilisateur. La visualisation permet de faire émerger une connaissance d'une manière similaire aux méthodes existantes citées précédemment tout en laissant à l'expert un contrôle total sur les données, et la possibilité de nettoyer cette connaissance. De nombreux accessoires peuvent ensuite être envisagés : consulter la bibliographie, accéder aux portails du domaine, annoter les données, etc.

[1] par exemple la longueur du plus court chemin entre deux nœuds ou le nombre d'ancêtres communs

Méthode de regroupement

Il existe de nombreuses méthodes de regroupement, et certains environnements spécifiques à cette tâche existent déjà. Weka est généraliste alors que MultiExperiment Viewer concerne la communauté bioinformatique. Les travaux menés au sein de notre laboratoire utilisent un algorithme d'analyse de données est basé sur la logique floue [Chiu 1997]. Le regroupement flou a la particularité d'associer un élément à plusieurs classes avec un degré d'appartenance à chacune. D'un point de vue général, cela permet de représenter un degré d'incertitude dans l'analyse. Dans le contexte de données biologiques, cela permet aussi de représenter qu'un même gène peut produire différentes protéines et ainsi participer à différentes fonctions. Ce besoin a par la suite été exprimé par Eisen qui, à partir d'un exemple concret, exprime le besoin de représenter des diagrammes d'Euler (figure 7.8 & figure 7.9).

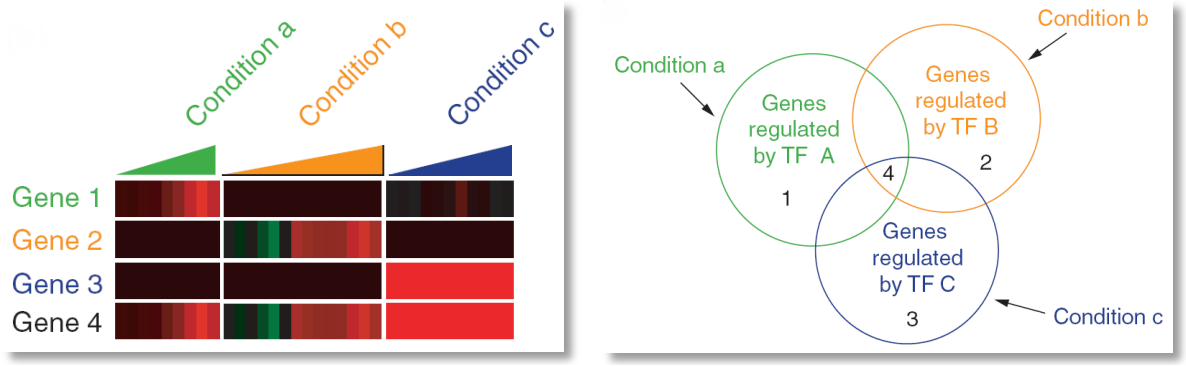


Figure 7.8 – Exemple explicatif proposé par Gasch et Eisen pour justifier le besoin de représenter l'intersection entre plusieurs ensembles de gènes. Figure issue de [Gasch and Eisen 2002].

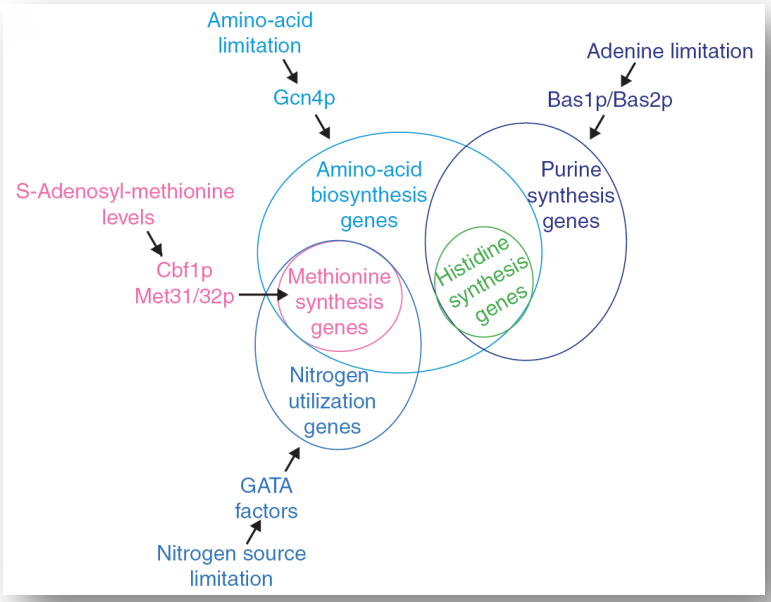


Figure 7.9 – Exemple concret proposé par Gasch et Eisen montrant le partage fonctionnel de certains ensembles et sous-ensembles de gènes. Figure issue de [Gasch and Eisen 2002].

Intégration de plusieurs regroupements

Comme nous l'avons déjà présenté, quand bien même nous faisons le choix de nous restreindre à une classification attribuant un gène à une classe unique, la visualisation d'intersection d'ensembles s'avère parfois indispensable. Les contraintes organisationnelles d'un laboratoire où d'une équipe sont telles que les expériences sont le plus souvent découpées en plusieurs étapes, réparties, ou progressives. Les jeux de données ne sont alors pas toujours

comparables. Dans le cas de notre collaboration avec Y. Cayre, nous avons, par exemple, constaté que les puces provenant d'un même lot ne contiennent pas toujours les mêmes « spots ». Le problème est bien plus vaste lorsque l'on souhaite cumuler les résultats de jeux de données précédents.

Dans tous les cas, lorsqu'un jeu de données est complet, on peut souhaiter utiliser un jeu antérieur à des fins de contrôle ou d'amélioration de la fiabilité. Il s'agit de capitaliser les résultats expérimentaux. Les données n'étant pas comparables, on peut choisir de comparer les regroupements éventuels.

L'expérience qui suit

Dans l'expérience qui suit, nous avons voulu évaluer la méthode de regroupement flou. Pour cela, nous nous sommes basés sur le jeu de données publié par [Bozdech, Llinás et al. 2003]. Nous avons comparé les regroupements manuels résultant de leur expertise au regroupement flou automatique en superposant les deux.

7.3.2 Résultats visuels

Dans les captures qui suivent, le couleur est indispensable à la compréhension. Les gènes, comme auparavant, sont de couleur rose. Les classes établies dans [Bozdech, Llinás et al. 2003] sont de couleur vertes. Les centroïdes élus par l'algorithme de regroupement flou, qui sont des gènes, sont les cercles roses. Le diamètre des cercles varie avec le nombre de gènes associés à chaque classe.

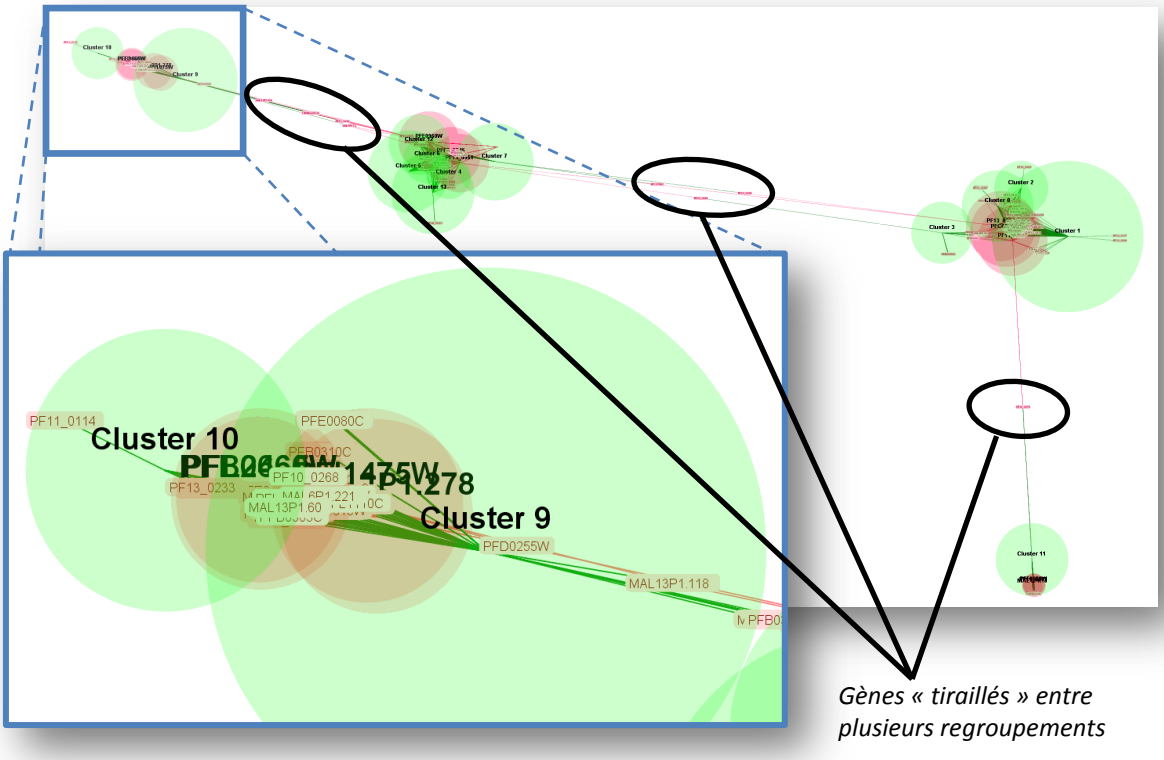


Figure 7.10 – Visualisation globale : les deux regroupements sont globalement cohérents, les éléments se recouvrent fortement et sont peu lisibles par conséquent.

Aperçu global puis zoom sur la partie supérieure gauche de la figure

La figure 7.10 présente une vue globale de l'analyse à l'initialisation de l'interface. Dans un premier temps, on constate une cohérence globale entre les résultats obtenus par Z. Bozdech et

le regroupement automatique dans 4 grandes régions de l'écran. On observe cependant la présence de quelques gènes « tiraillés » entre ces régions.

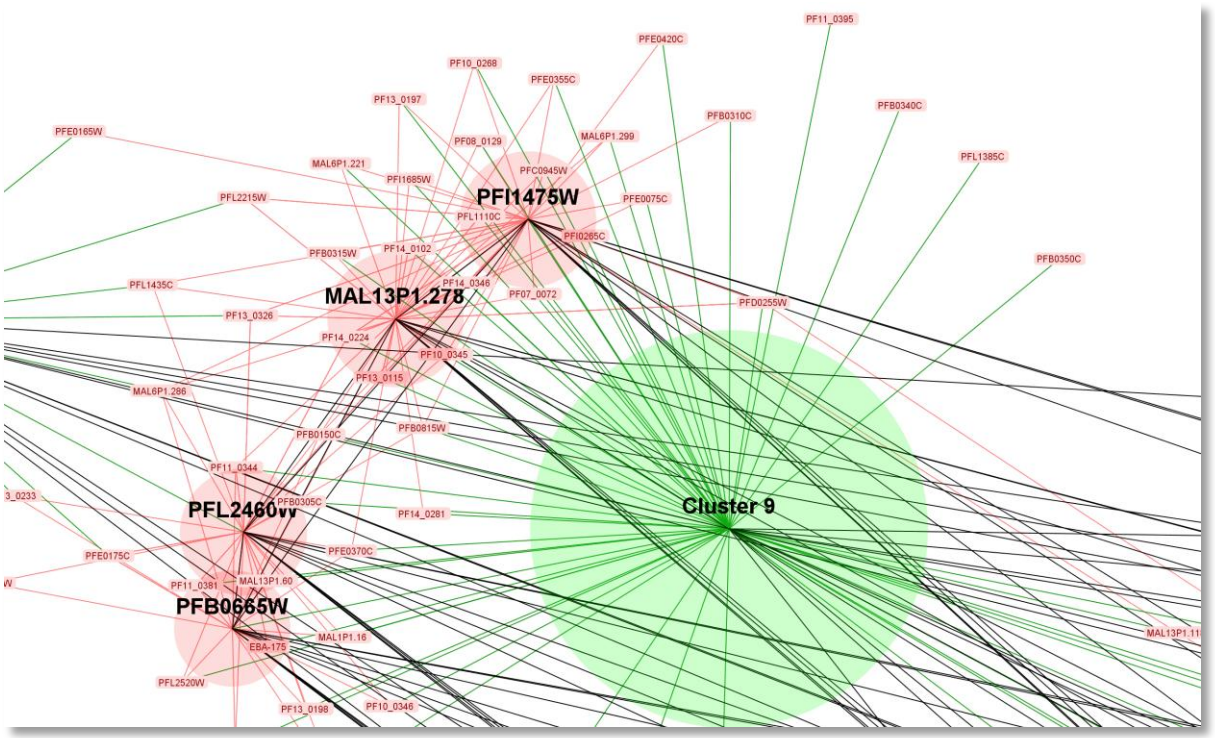


Figure 7.11 – Dans cette capture agrandissant la région supérieure gauche de la vue générale, les arêtes occasionnant la répulsion entre les nœuds sont visibles. La NBodyForce quant à elle ne donne pas lieu à la génération d'arêtes, il est donc impossible de spécifier une longueur ou une cible particulière. Elle se restreint à éviter le recouvrement de nœuds.

Même si on zoome, les nœuds dans cette visualisation se superposent pour la plupart, et le résultat pour être lisible nécessite de nombreuses manipulations à l'aide de la souris. Comme nous l'avons mentionné dans la section 6.3.2.1 (page 169), nous avons souhaité mieux espacer les éléments du graphe dans le plan. Le premier outil (NBodyForce [Barnes and Hut 1986]) créé une répulsion entre chaque élément et ses plus proches voisins les. Le second vise à éviter la superposition de groupes de gènes de mêmes types. D'un point de vue graphique, un cercle vert et un cercle rose peuvent se superposer, mais deux cercles roses ne peuvent se superposer, deux cercles verts ne peuvent pas non plus. Pour y arriver, nous avons simplement inséré des arêtes non visualisées (cf. figure 7.11). Leur longueur est calculée à partir de la somme des rayons des cerces qui sont aux extrémités ; l'élasticité de ces arêtes est particulièrement faible (rigide). Le résultat produit est présent dans la figure 7.12.

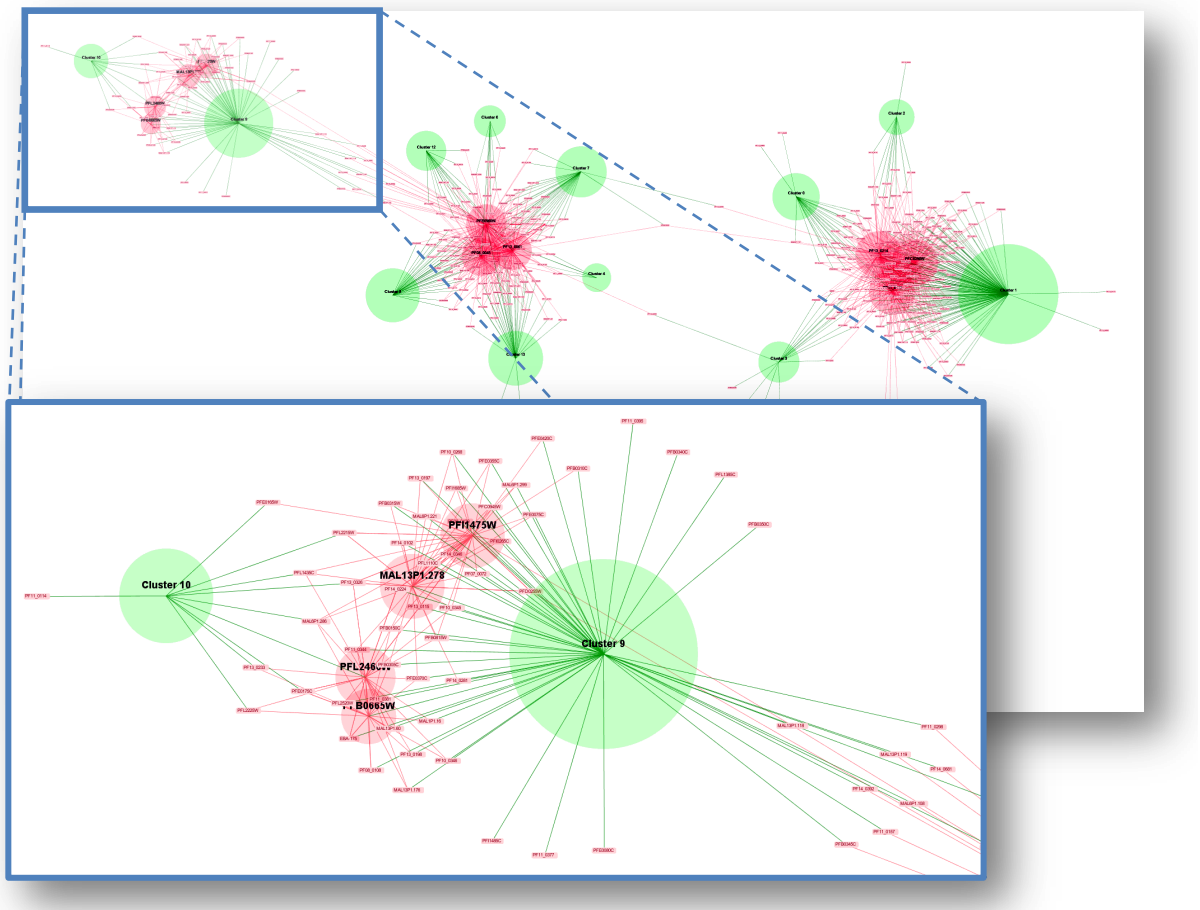
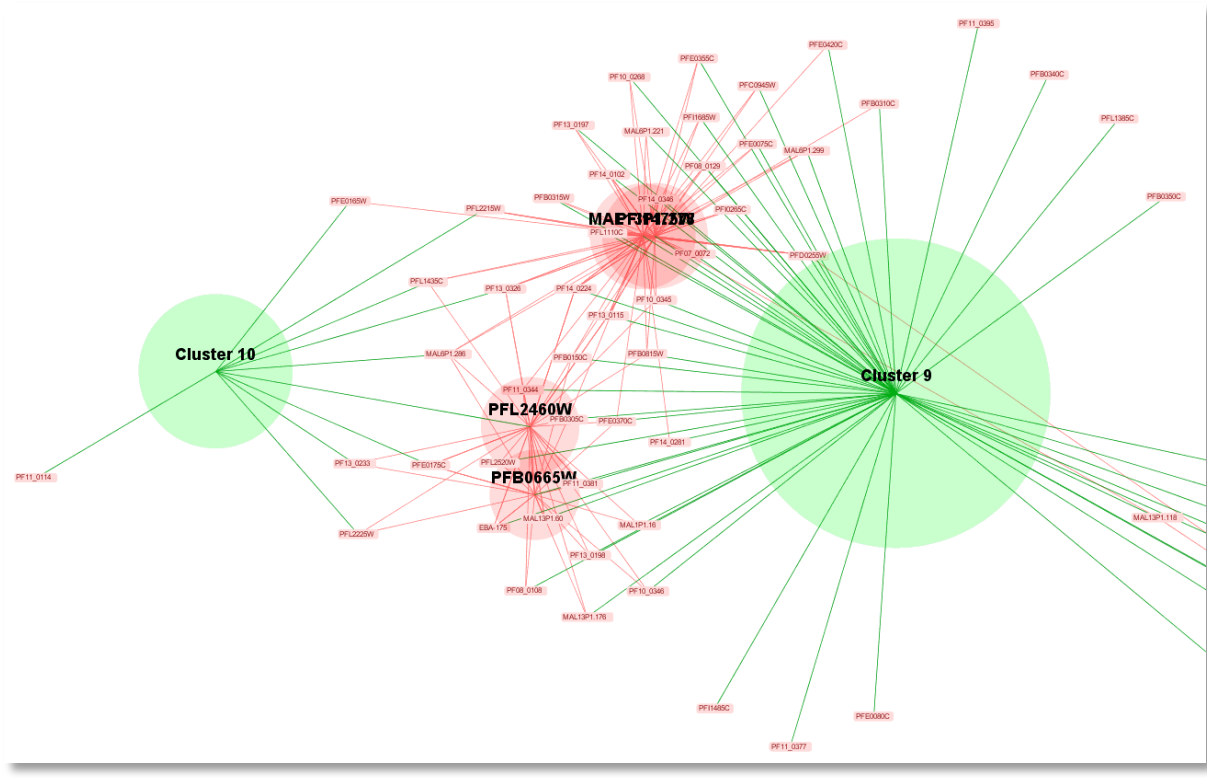


Figure 7.12 – Cette vue globale met en œuvre une séparation des groupes de gènes et une répulsion entre voisins proches.

Le constat que nous dressons est dual vis-à-vis du précédent : le graphe occupe mieux le plan, les nœuds ne se superposent pas et sont visibles. Les déformations qui produisent ce résultat sont cependant source d'une distorsion importante du point de vue des distances. Ces distances ne sont parfois plus significatives. Les nœuds attachés à une classe de Z. Bozdech ont une distance constante avec leur classe. En évitant la superposition des nœuds, certains sont placés plus proches du centre que d'autres. Lorsqu'il s'agit de regroupement flou, le même phénomène peut apparaître : un gène avec un degré d'appartenance supérieur à un voisin peut être plus éloigné malgré tout.

On peut alors faire le choix de désactiver la force de répulsion généralisée (NBodyForce) ou la répulsion entre regroupements de mêmes types. La figure 7.13 montre le résultat obtenu si on autorise la superposition de groupements de même type tout en conservant la force de répulsion généralisée. On constate que deux cercles rouges se superposent malgré la répulsion qui les concerne. Cela résulte du fait qu'ils partagent un grand nombre de gènes en commun : les forces qui les relient à leurs gènes sont plus fortes que la répulsion et les oblige à converger en un même point.

Du point de vue métier, cela peut inciter à faire le choix de fusionner ces deux centroïdes en une seule classe, ou à augmenter le nombre de centroïdes dans cette région et relancer un regroupement flou avec de nouveaux paramètres.



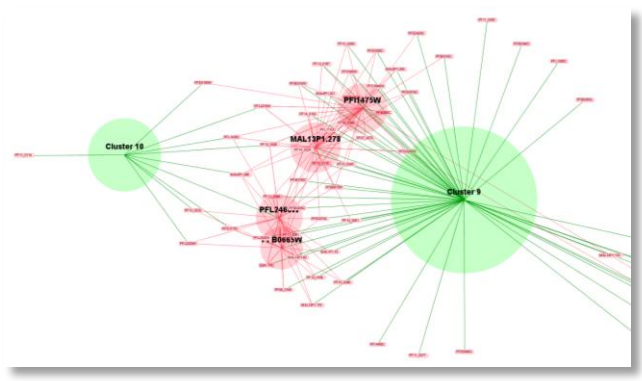


Figure 7.13 – La capture ci-dessus met en œuvre la répulsion entre voisins, mais ne permet pas d'éviter le recouvrement des classes. Bien au contraire, le fait que deux groupes rouges partagent un grand nombre de gènes communs les contraint à se superposer malgré la force de répulsion générale qui s'applique à eux. La figure gauche montre le résultat obtenu avec les arêtes évitant le recouvrement de classes, déjà présenté dans la figure 7.12.

La figure 7.14 présente la seconde alternative consistant à désactiver la répulsion généralisée, mais à conserver la force interdisant la superposition de deux classes de même type. Une troisième visualisation possible consiste à désactiver la répulsion généralisée, en conservant l'interdiction de superposition des ensembles. Cette alternative permet de mieux dissocier certains amas de gènes et les groupes auxquels ils appartiennent tout en respectant mieux les distances et donc leurs degrés d'appartenance.

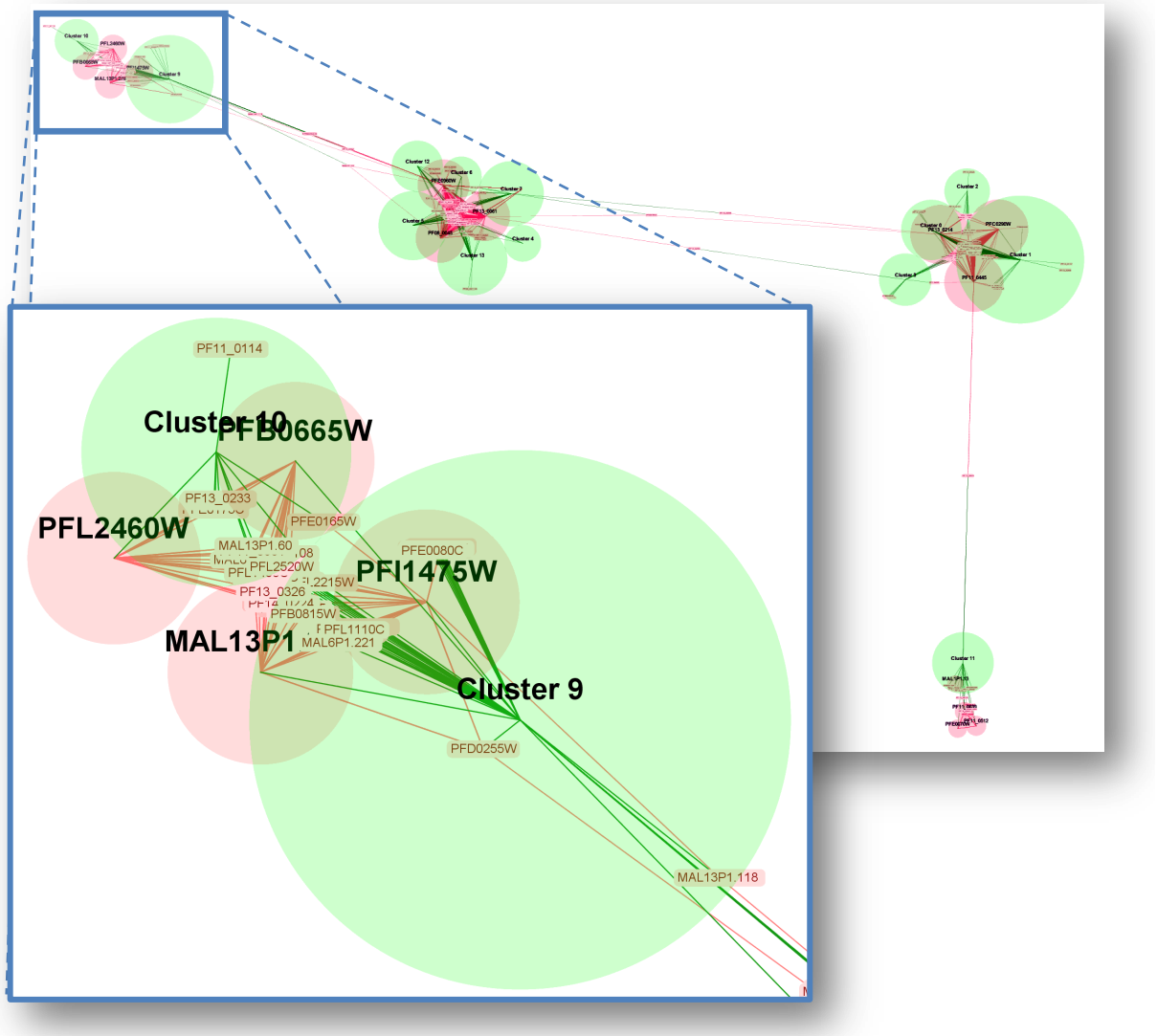


Figure 7.14 – Dans cette troisième vue globale, la répulsion de proximité est désactivée, mais les regroupements de mêmes types ne peuvent se superposer.

Positionnement manuel des nœuds

Comme nous l'avons indiqué, il est possible d'effectuer certains réglages au niveau d'un type ou d'une instance. Il est ainsi possible de fixer un ou plusieurs éléments dans le plan. Nous souhaitons présenter ici l'intérêt de cette plasticité de l'interface. La figure 7.15 montre la même région que précédemment alors qu'aucune répulsion n'est mise en œuvre. Le résultat est illisible : tous les gènes liés à un centroïde flou sont amalgamés entre les deux classes produites par Z. Bozdech.

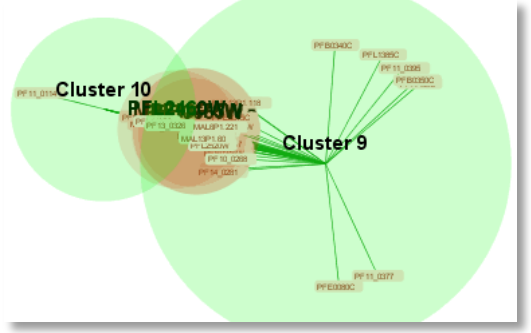


Figure 7.15 – Lorsqu'aucune répulsion n'est active, les gènes et regroupement flou ont tendance à s'amalgamé au centre.

La solution précédente consiste à écarter automatiquement les éléments graphiques entre eux. L'utilisateur peut le réaliser manuellement en fixant certaines classes par exemples et en les écartant. Les résultats sont présentés dans les figures suivantes. La figure 7.16 et la figure 7.17 montre l'ensemble de la région déployée. La force de répulsion généralisée est active dans la première, inactive dans la seconde.

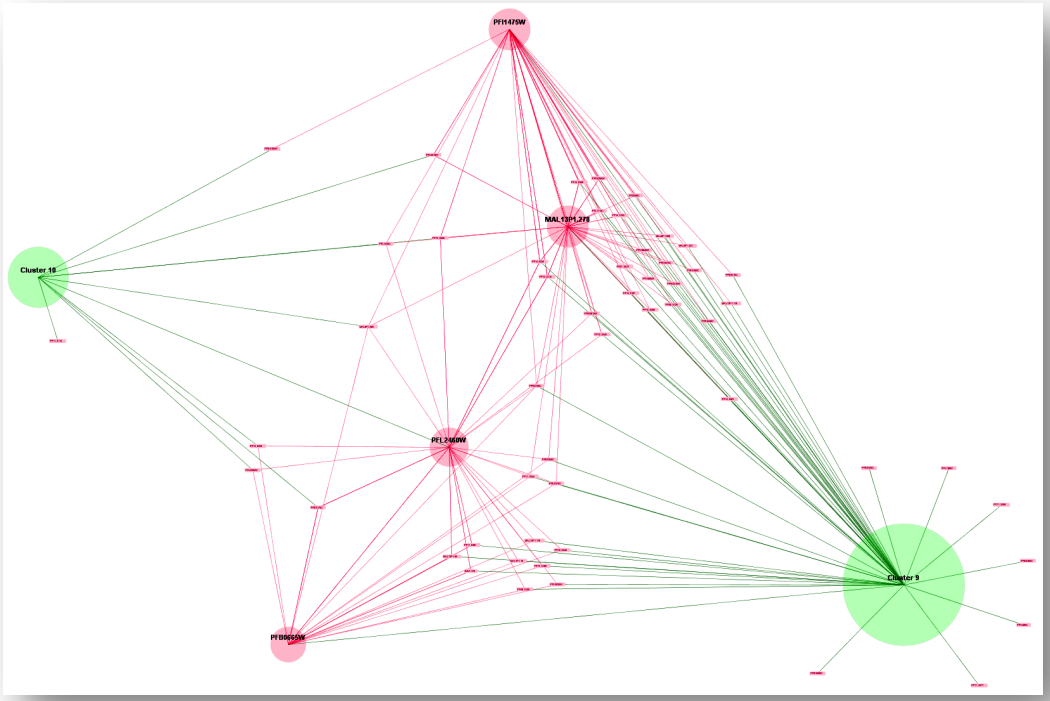


Figure 7.16 – Disposition fixée par l'utilisateur avec répulsion généralisée.

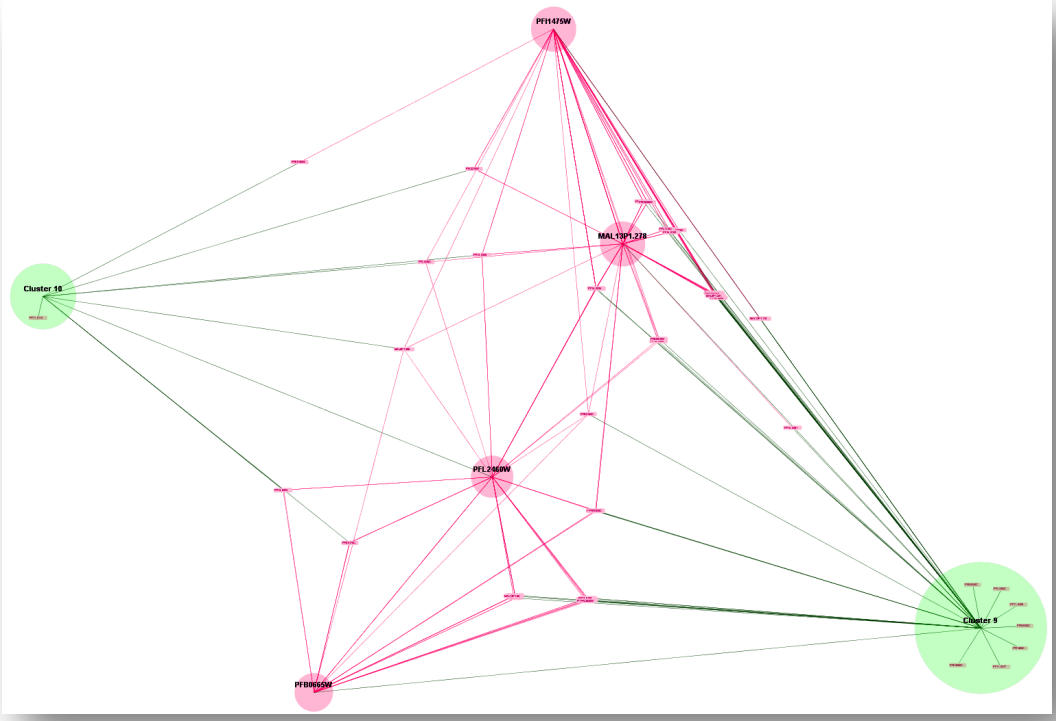


Figure 7.17 – Disposition fixée par l'utilisateur sans répulsion généralisée.

Cette visualisation semble dans un premier temps exagérée ; elle peut apparaître sans intérêt pour certains car les distances n'ont que peu de lien avec celles présentes dans l'espace d'origine. Cependant, les tractions exercées sur les gènes sont plus fortes. Grâce à cela, les gènes précédemment amalgamés sont plus finement présentés et on distingue mieux l'association entre un sous ensemble de gènes et ses différentes classes. Si les valeurs de distance ne sont pas respectées, les nuances entre les différents degrés d'appartenance sont mieux mises en évidences. La figure 7.18 agrandit une région de la dernière capture et présente ainsi la finesse dans le voisinage d'un des ensembles flous.

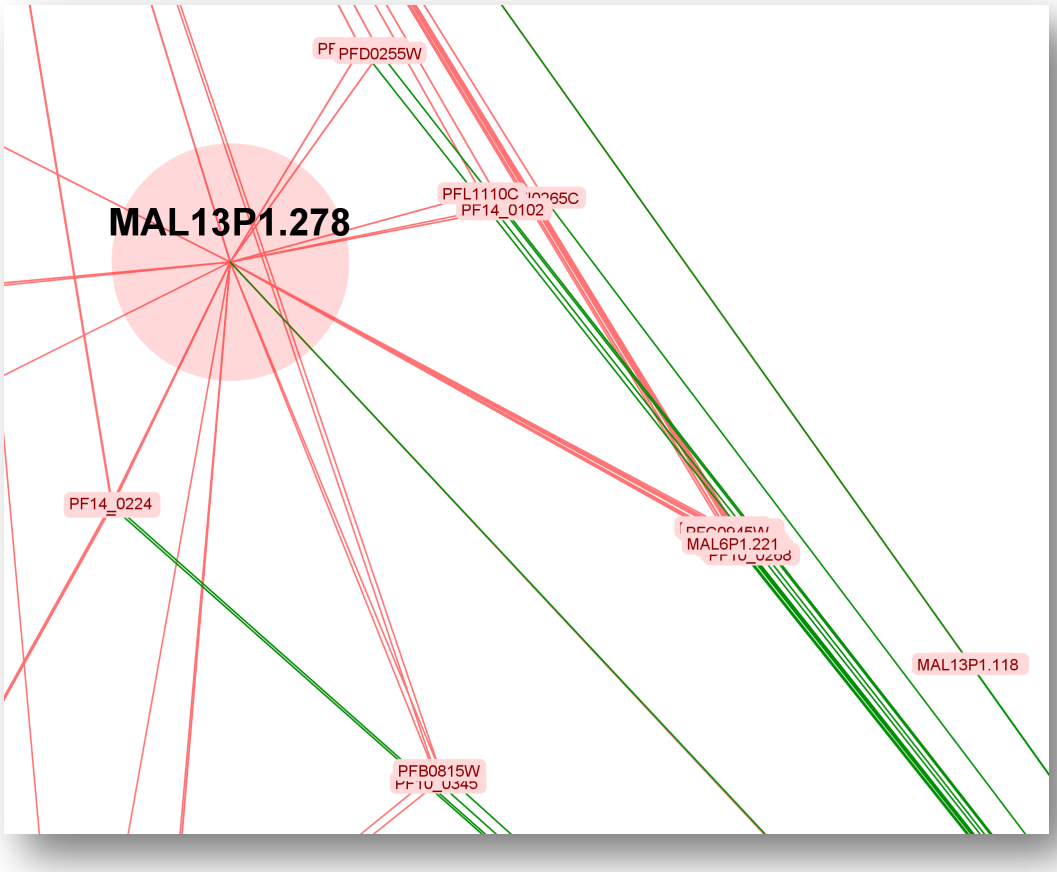


Figure 7.18 – Agrandissement du voisinage d'un des ensembles flous. Les sous-ensembles de gènes forment des petits groupes plus finement distinguables.

Exemple de fonctionnalité spécifique

Nous avons présenté dans la section 6.3.2.2 un composant appelé BiSlider. Nous l'avons mis en œuvre dans le cadre de l'analyse de données d'expression. Couplé à un filtre, il permet de gérer un seuil minimum et maximum dans le degré d'appartenance. Lorsque ce degré d'appartenance d'un élément à un ensemble flou est en dehors de l'intervalle défini, l'arête est éliminée de la vue et la force ne s'applique plus. Les vues suivantes montrent le résultat obtenu lorsque l'on ne conserve que des valeurs supérieures, respectivement, à 0,3, 0,5, et 0,8.

Faire varier ces valeurs est important : de façon immédiate, cela permet d'évaluer la qualité ou la fiabilité du résultat de l'analyse. De plus, les réglages de regroupements flous sont généralement difficiles à obtenir et à évaluer. Le résultat d'un regroupement flou est une matrice associant chaque gène à chaque centroïde au travers d'un degré d'appartenance. Cela produit un graphe particulièrement dense (potentiellement complet). Filtrer les valeurs permet de s'adapter à la distribution des valeurs d'appartenance. On fait évoluer le paysage en ne conservant que les associations les plus fortes et donc *a priori* les plus fiables.

Dans l'exemple qui suit, cela permet par exemple de relativiser les contradictions entre les groupes définis par Bozdech et ceux obtenus automatiquement. Lorsque l'on filtre avec un degré d'appartenance élevé, certains contradictions disparaissent, et d'autres persistent. Celles qui persistent sont probablement plus pertinentes. Plus généralement, c'est toute la connexité du graphe qui évolue avec l'amoindrissement de sa densité.

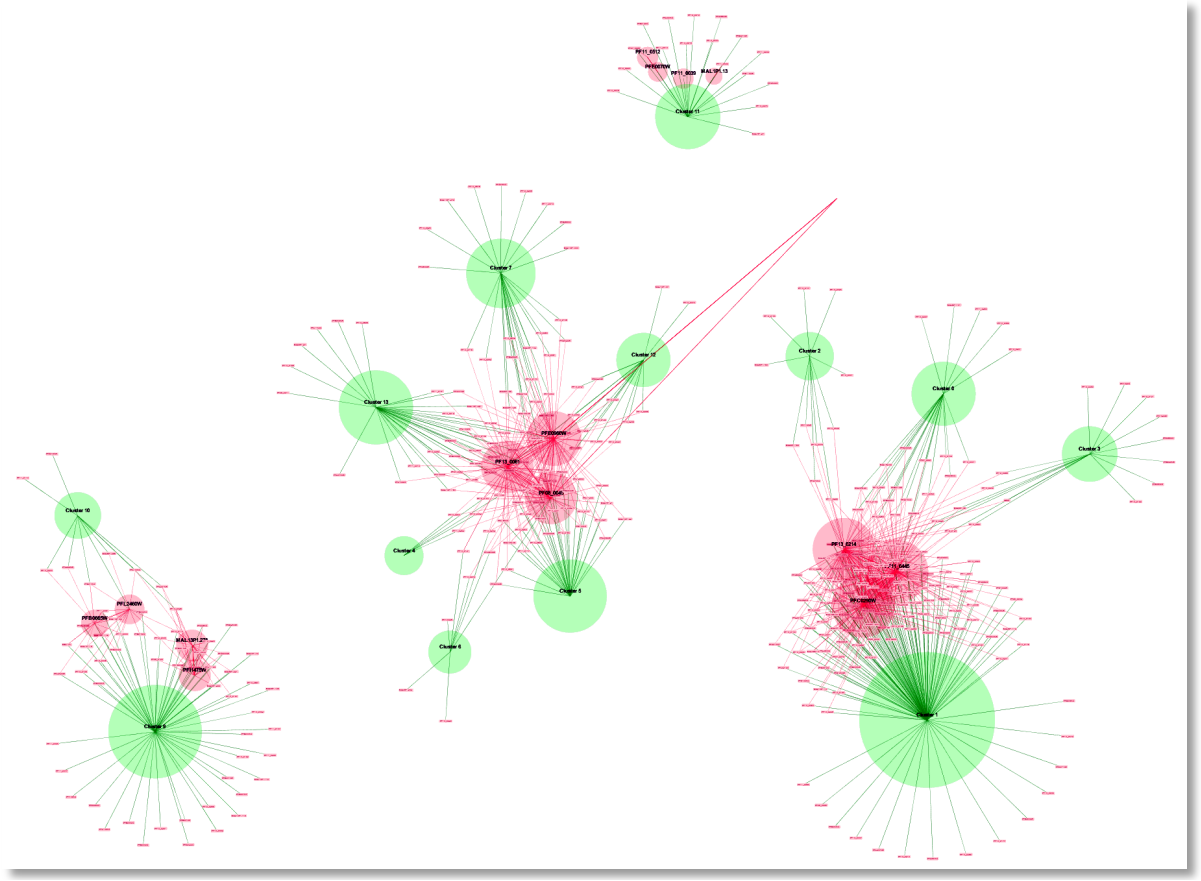


Figure 7.19 – Graphe obtenu avec un degré d'appartenance minimum de 0,3.

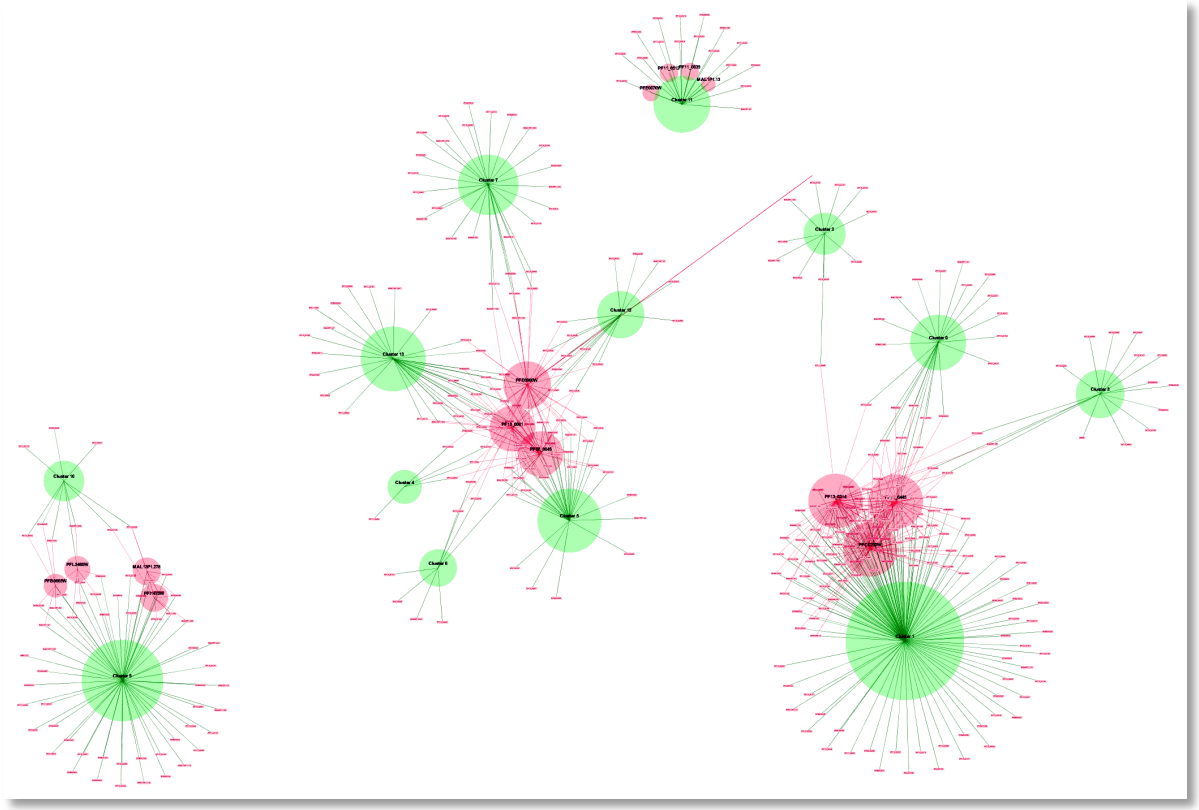


Figure 7.20 – Graphe obtenu avec un degré d'appartenance minimum de 0,5.

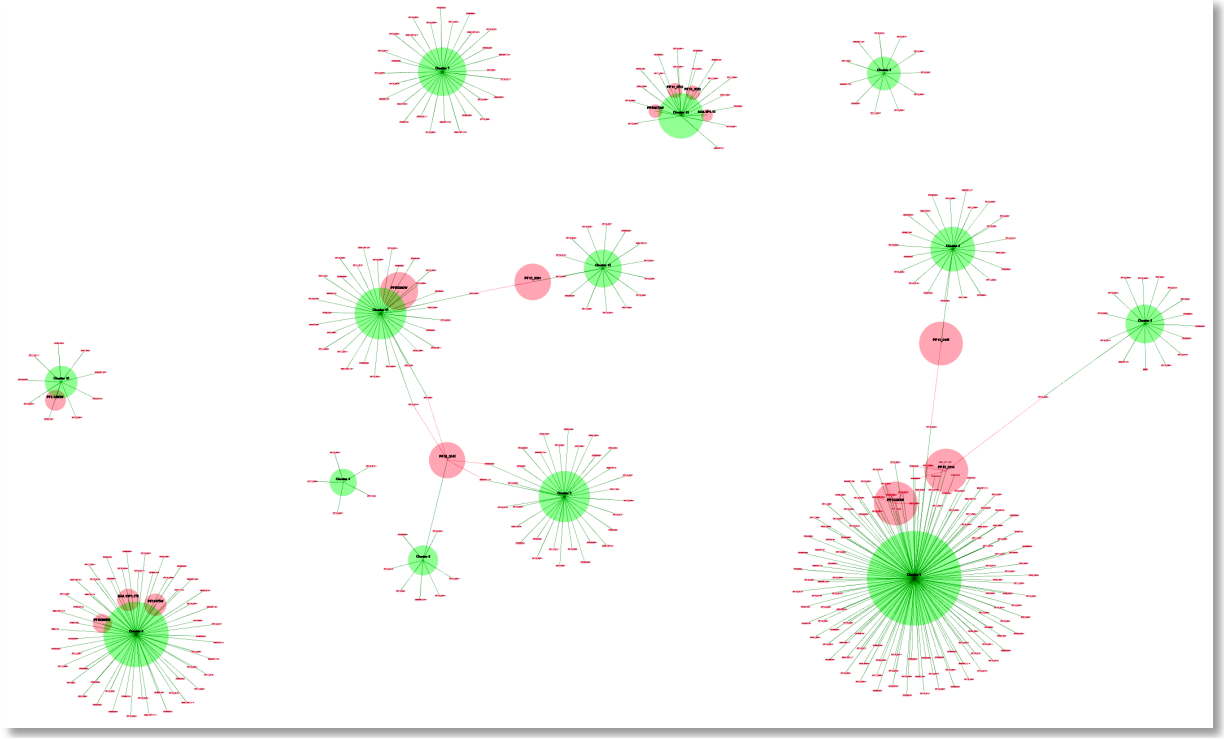


Figure 7.21 – Graphe obtenu avec un degré d'appartenance minimum de 0,8.

Lentilles métier

Nous avons déjà présenté les mécanismes de lentilles métier. Nous présentons ici les résultats produits par celle décrite précédemment dans ce document. Elle consiste, pour un ensemble donné (cercle vert ou rouge) à foncer les gènes associés, changer leur étiquette en affichant le résumé provenant des bases du domaine, et enfin à disposer autour de cela les concepts correspondant aux annotations. La figure 7.22 illustre ces propos.

Si cette lentille ne permet pas immédiatement de faire apparaître un terme unique, on peut rapidement constater une corrélation entre les fonctions connues des gènes et les annotations. Ces informations sont fortement cohérentes et permettent rapidement de déterminer les contours fonctionnels du « cluster 10 ». Si cette visualisation nécessite un effort minimum de la part de l'utilisateur en comparaison d'autres méthodes totalement automatisées il peut ici conserver un contrôle : supprimer des éléments, les déplacer, les catégoriser, les figer, etc. L'impact est immédiat et continue sur la vue des données. S'il y a une incohérence, ou un phénomène remarquable mais isolé, il peut s'en apercevoir assez facilement et se concentrer dessus. Il dispose alors immédiatement de mécanismes comme l'ouverture du portail du domaine sur le gène donné, etc.

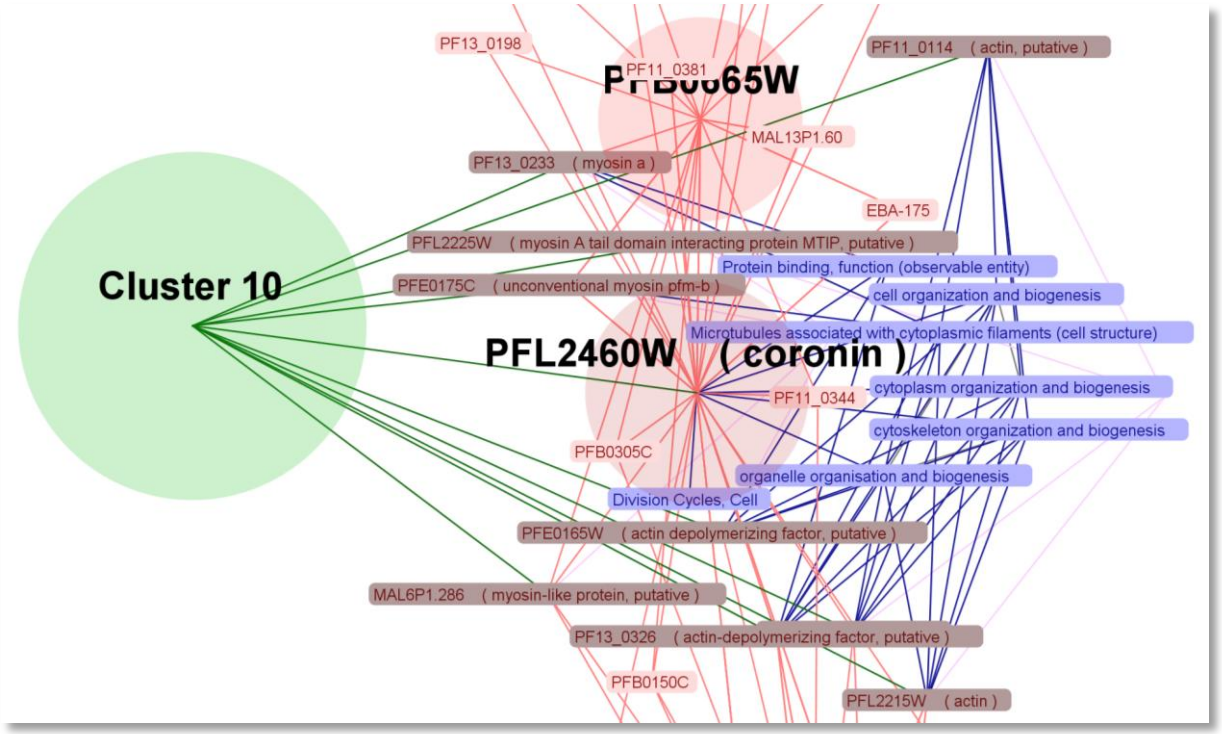


Figure 7.22 – Résultat obtenu par une lentille métier. Le « cluster 10 » est sélectionné : les gènes associés sont foncés et leur résumé est affiché dans l'étiquette. Les annotations (concepts en bleu) sont disposées autour.

Lentille générique de recherche

Nous avons développé une seconde lentille associée à une boîte de dialogue spécifique. Elle permet une recherche basée sur un motif textuel (une expression régulière). Dans la pratique, si on rentre une chaîne de caractère dans cette boîte de dialogue, les éléments dont le nom contient cette chaîne sont mis en évidence. De la même façon que précédemment, on peut découper ce système en plusieurs éléments :

- la lentille qui permet de filtrer et modifier l'ensemble d'éléments,
- la boîte de dialogue qui permet de rechercher les éléments répondant au motif et qui est associée à la lentille.

La lentille de recherche permet d'associer à une structure de données (la sélection) un modifieur. Dans l'exemple présenté dans la figure 7.23, les éléments sont rendus visibles et sont foncés. La boîte de dialogue récupère la taille du résultat de la recherche en accédant directement à la sélection. Ici, 66 gènes répondent à la requête.

D'autres actions sont envisageables. Par exemple, il est simple de demander de centrer la vue sur l'élément qui répond à la requête, s'il est unique. Si plusieurs éléments répondent à la recherche, on peut proposer d'ajouter la position et l'échelle de la vue automatiquement. On peut enfin considérer cette sélection comme sélection principale et affecter les opérations des menus contextuels. Cette lentille est générique et n'a rien de spécifique à l'analyse de données d'expression.

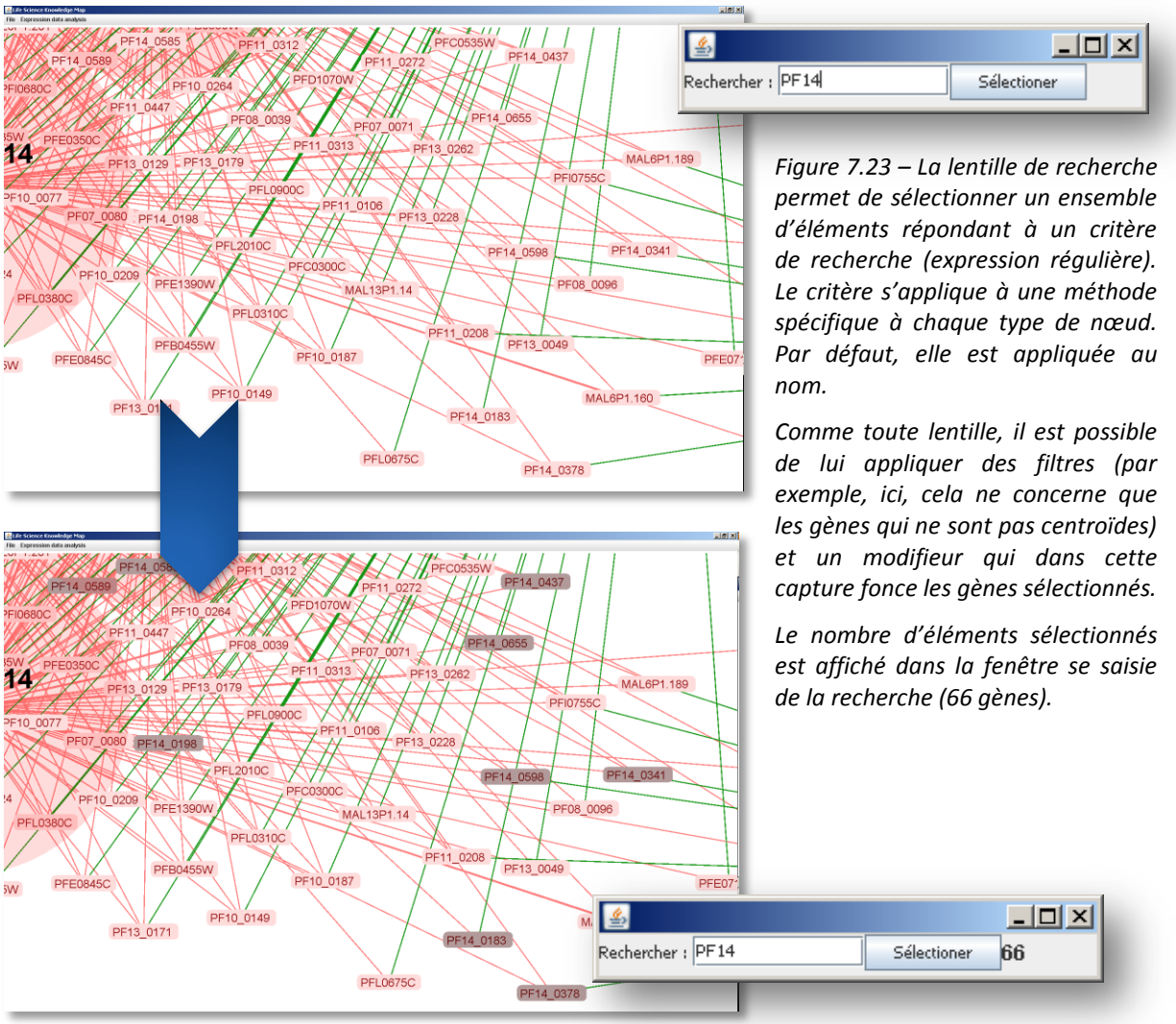


Figure 7.23 – La lentille de recherche permet de sélectionner un ensemble d'éléments répondant à un critère de recherche (expression régulière). Le critère s'applique à une méthode spécifique à chaque type de nœud. Par défaut, elle est appliquée au nom.

Comme toute lentille, il est possible de lui appliquer des filtres (par exemple, ici, cela ne concerne que les gènes qui ne sont pas centroïdes) et un modifieur qui dans cette capture fonce les gènes sélectionnés.

Le nombre d'éléments sélectionnés est affiché dans la fenêtre se saisie de la recherche (66 gènes).

7.3.3 Bilan

Nous venons de présenter deux applications pour deux tâches spécifiques, complexes et qui diffèrent fortement : la conception d'une RTO et l'analyse de données d'expression. Les utilisateurs diffèrent, les données aussi, les opérations métiers, les fonctionnalités, etc. Pourtant, I²DEE répond à un ensemble de besoins communs en permettant de générer deux cartes adaptées à ces contextes hétérogènes et aux sous-tâches qui composent l'activité de l'utilisateur.

Les captures que nous avons présentées dans ce chapitre ont mis en évidence cette multiplicité des besoins et ont montré comment la plasticité de l'interface permet de s'y adapter. I²DEE est notamment capable de gérer des topologies multiples et des quantités de données importantes à différente échelles.

7.4 Synthèse et discussions

Nous venons de présenter les résultats visuels obtenus à partir de nombreuses captures d'écran. Les différents exemples nous ont permis de voir l'intérêt pour l'utilisateur que l'interface s'adapte à la tâche en cours. Toutes ces fonctionnalités sont utiles : activer certaines forces, figer certains nœuds, utiliser des types d'éléments distincts de ceux de l'entrepôt, spécifier des lentilles métier, etc. La topologie des données est variable. Une tâche complexe réalisée par un expert peut souvent se décomposer en sous-tâches qui nécessitent des besoins plus spécifiques en termes de visualisation.

Une limite : comment gérer la superposition des arêtes ?

Du point de vue technique, une limite importante n'a pas été décrite. L'entrepôt permet de stocker plusieurs arêtes reliant deux mêmes nœuds. Le dessin des arêtes est pour l'instant rectiligne. Actuellement, la boîte à outils gère cela en instaurant des niveaux dessinés successivement. Il est ainsi possible de spécifier que les arêtes issues de regroupements manuels de Bozdech recouvrent ou soit masquées par celle arêtes provenant de regroupements flous. Cependant, l'information affichée n'est pas complète. Le dessin non rectiligne des arêtes permet d'afficher les arêtes sans qu'elles se superposent, mais le résultat visuel devient trop complexe et difficilement interprétable. Pour fournir à l'utilisateur une information complète, nous proposons pour l'instant des panneaux d'information génériques. Un premier panneau d'information permet, après la sélection d'un nœud par un click, d'afficher les informations qui lui sont relatives dans deux colonnes :

- les attributs sont placés dans la première colonne et leurs valeurs dans la secondes,
- les types d'arêtes et valeurs sont placés dans la première colonne, les éléments associés dans la seconde.

Un second panneau, similaire, apparait à la sélection d'une relation et décrit toutes les arêtes qui se superposent sur le point cliqué. Ces mécanismes ne sont pas suffisants, trop complexes et peu visuels. Nous prévoyons donc dans un second temps de combiner ces différentes solutions et d'introduire un nouveau paramétrage permettant de choisir la méthode employée pour un type de relation : dessin non rectiligne et présence dans les panneaux d'information. A termes, des méthodes complémentaires et mieux adaptées doivent être recherchées.

Evaluation par des utilisateurs biologistes

Les discussions précédentes présentent les capacités d'I^{2}DEE essentiellement d'un point de vue théorique et technique. Nous avons réalisé une évaluation avec deux biologistes dans le cadre des deux applications. Plusieurs constats ont été observés.

Le premier est que les deux utilisateurs ont fortement associé les gènes par rapport à leurs distances et non à leurs connexions. Ce résultat nous étonne dans la mesure où il contredit ceux obtenus par Mackinlay [Mackinlay 1986]. Le second est celui d'une difficulté à manipuler cette nouvelle visualisation. Certains réglages font défaut et les expériences étaient assez courtes (autour de 20 à 30 minutes de manipulation). A la fin de l'expérience, des progrès sensibles avaient été réalisés dans la manipulation. On peut ainsi envisager après amélioration de la visualisation que l'utilisateur soit familier de l'environnement après quelques heures de manipulation seulement. Cet investissement de la part de l'utilisateur est d'autant plus envisageable qu'il est potentiellement commun à de multiples outils basés sur I^{2}DEE.

Dans le contexte de l'ingénierie des connaissances, ces expériences ont révélé un manque de familiarisation du biologiste avec des outils d'ingénierie des connaissances. Ils se focalisent constamment sur la connaissance biologique, et non sur les concepts qui la structurent. Bien qu'expert du domaine, le biologiste peut rencontrer des difficultés à comprendre la structure de la RTO proposée et les liens entre concepts. Très souvent, au travers d'une relation entre deux termes, c'est une relation biologique qui est recherchée, et en priorité à partir des distances et non des arêtes. De plus, contrairement à un tableau de données d'expression, l'aspect graphique

de l'interface induit chez l'utilisateur une volonté de retrouver un niveau formel et plus fiable dans l'interface. Il cherche ainsi des explicitations biologiques pour la cooccurrence, l'hyperonymie, etc.

Nous en concluons que dans les applications clientes telles que l'analyse de données d'expression de gènes, l'affichage des relations sémantique n'est pas nécessaire. Il est préférable de proposer des mécanismes d'initialisation regroupant des termes voisins. Dans les graphes denses, l'auto-organisation des concepts provoque des superpositions qui ne correspondent pas à une réalité des données et résulte d'aléas de projection de nombreuses contraintes dans deux dimensions. Il est ainsi particulièrement important de pouvoir laisser à l'utilisateur la possibilité de réduire la densité du graphe autant que possible, de filtrer les données, et d'adapter la vue en conséquence.

Les retours sont globalement très positifs. L'utilisateur voit un intérêt à terme dans l'outil, tenant compte de l'état actuel de prototype expérimental. Par ailleurs, des observations particulières nous encouragent : lorsque l'utilisateur s'est focalisé sur les relations de cooccurrence dans l'expérience d'ingénierie ontologique, il a constaté une corrélation avec la réalité biologique. L'utilisateur a lui-même été dans la mesure de percevoir et d'exprimer les difficultés propres au jeu de données initial qui concerne *Plasmodium Falciparum*. Les imperfections de la carte reflètent l'état des connaissances sur cet organisme : celles-ci sont pauvres et semblent se rapprocher de celles concernant les bactéries plutôt que de celles qui concernent les eucaryotes.

CHAPITRE 8 Synthèse & Conclusion

« Solving a problem simply means representing it so as to make the solution transparent. If the problem solving could actually be organized in these terms, the issue of representation would indeed become central. But even if it cannot – if this is too exaggerated a view – a deeper understanding of how representations are created and how they contribute to the solution of problems will become an essential component in the future of theory of design. »

SIMON, 1981

8.1 Introduction

Le principal objectif de cette thèse est de répondre au problème du partage, de l'accès et du traitement de l'information pour un utilisateur biologiste. I²DEE est le prototype qui en découle. Dans ce dernier chapitre, nous rappelons les caractéristiques clés de notre approche, leurs conséquences, leurs avantages et leurs limites. Pour cela nous abordons dans un premier temps les aspects architecturaux et techniques, avant de considérer nos résultats du point de vue de l'utilisateur final.

Notre travail de recherche nous a par ailleurs ouverts un grand nombre de directions à poursuivre. La version actuelle d'I²DEE implémente diverses fonctionnalités centrales à l'environnement, et d'autres, plus annexes, permettant de montrer l'efficacité de notre approche. De nombreuses considérations ne sont pas implémentées et nécessitent de l'être. La seconde partie de ce chapitre présente donc nos principales perspectives à court et long terme.

8.2 Synthèse des caractéristiques de notre approche

Dans cette section, nous proposons une synthèse des éléments caractéristiques de notre approche. Nous nous focalisons, dans un premier temps, sur les aspects techniques et architecturaux qui procurent souplesse, extensibilité et ouverture. Dans un second temps, ce sont les aspects fonctionnels et les interactions de l'utilisateur final qui sont mis en évidence.

8.2.1 Aspects architecturaux et techniques

La taxonomie du domaine de l'intégration qualifie le plus souvent les systèmes d'intégration suivant des termes spécifiques. Ce que nous avons recherché, en quelque sorte, est d'éviter tout cloisonnement ou de n'en ajouter aucun, mais au contraire de nous ouvrir à l'ensemble des approches existantes. Se spécialiser, c'est avant tout contraindre le développeur et l'utilisateur, les opposer dans leurs besoins. C'est donc limiter l'utilité, l'utilisation et la réutilisation.

Niveau d'intégration

Nous sommes restés dans une approche générique, qui se veut ouverte à tous les domaines. Notre architecture est capable de répondre à des besoins techniques divers (système d'intégration, langage de requête, service Web, etc.). Pour cela, notre modèle est en réalité un métamodèle. Un degré d'abstraction supplémentaire nous permet d'éviter les problèmes de réconciliation des schémas. Cette approche est utilisée par certains systèmes d'intégration médiateurs notamment qui ne peuvent matérialiser les données et doivent tout réaliser entièrement automatiquement. Si les conflits sont moins nombreux, il est en revanche plus complexe d'exprimer des contraintes sur le schéma. Cependant, nous avons fait le choix de matérialiser les données. I²DEE peut ainsi être qualifié d'entrepôt. N'ayant pas à gérer l'intégration des schémas et des instances dynamiquement, notre métamodèle peut s'avérer plus simple, et donc extensible et souple.

La description du métamodèle est simple, si bien qu'on peut assimiler notre approche à une approche lâche de l'intégration. Mais la prise en compte de multiples ontologies alignées au sein d'un médiateur (initialement UMLS) permet de réaliser une intégration sémantique. Le domaine de l'intégration définit habituellement l'intégration sémantique comme celle réalisée au niveau des instances (intégration verticale, cf. section 2.3.1.3 page 59). Nous qualifions ici notre intégration de sémantique elle car elle se place à un niveau d'interopérabilité sémantique : nous avons la possibilité de prendre en compte la polysémie (un terme associé à plusieurs concepts), la synonymie (deux termes associés au même concept) et le multilinguisme.

Approche matérialisée

Notre approche se caractérise concrètement par un système d'intégration : les données sont intégrées et interrogeables via un langage de requête. Il s'agit d'une approche matérialisées et donc d'un entrepôt. Cependant, comme nous l'avons introduit au début de cette section, nous ne souhaitons pas restreindre les usages, du point de vue du développeur comme de celui de l'utilisateur final. Le métamodèle nous permet de :

- concevoir facilement des vues et donc de répondre aux contraintes de schémas d'autres systèmes d'intégration, entrepôts ou médiateurs. La matérialisation des vues proposées par le SGBD permet de s'abstraire partiellement des problèmes de performances liés à l'adaptation entre le modèle et la vue.
- générer des services Web automatiquement et être utilisé par des gestionnaires de flux de travail ou de données,
- proposer un portail Web générique,
- concevoir des clients graphiques lourds (plateformes) grâce à une boîte à outils,
- réaliser des requêtes sous forme de chemins et être utilisé comme système à base de liens ou de chemins (navigateur).

Si plusieurs systèmes cohabitent et exploitent I^2DEE, la donnée est au cœur du système. C'est au travers de ces données que l'interopérabilité des systèmes utilisateurs est assurée. Par exemple, les annotations réalisées directement sur les données au travers d'un client lourd sont directement répercutées sur le portail et réciproquement. De même, l'ajout de documents réalisé dans un client peut être répercuté dans d'autres clients, pourvu que cela se matérialise dans l'entrepôt (ou dans le ficher) par l'ajout d'un élément.

Approche distribuée

Notre approche se veut large, adaptée à de nombreux contextes. Pour autant, nous n'envisageons pas uniquement des entrepôts faisant autorité et peu nombreux. Au contraire, la réalité du besoin des laboratoires et des équipes de recherche impose que chaque entité qui le souhaite puisse être propriétaire d'un entrepôt. L'interopérabilité des différentes ressources repose principalement sur trois éléments de synchronisation et de standardisation :

- une gestion commune des identifiants,
- une gestion commune des métadonnées (types, noms de sources, noms de propriétés, etc.),
- la présence dans un entrepôt d'une historisation des modifications des données.

Concernant la gestion commune des identifiants, LSID est un standard qui s'avère potentiellement un bon candidat. Plus généralement, la communauté biomédicale a déjà réalisé un effort considérable de structuration. PubMed recense la littérature, NAR des outils bioinformatiques (bases de données, applications et services Web, etc.), et de grands *consortiums* regroupent la majeure partie des données de séquences, en génomique et en protéomique. L'hypothèse d'une telle ressource centralisant la gestion de l'identification et des métadonnées est donc réaliste. La dernière condition concernant l'historisation des modifications des données peut être encapsulée dans l'entrepôt et être transparente pour le développeur et l'utilisateur.

Des supports multiples

Au-delà d'une approche distribuée, nous pensons qu'il est nécessaire de proposer des supports multiples. Il est ainsi possible de sauvegarder les données d'une carte en XML indépendamment ou non de sa description graphique. Il est aussi possible d'exporter la carte visualisée via notre boîte à outils directement dans un format d'image courant (PNG, JPG, etc.) et dans une haute résolution (la limite provient uniquement de la capacité de mémoire de la machine virtuelle). Nous envisageons d'autres supports tels que des fichiers formatés pour des tableurs.

Une modèle relationnel et une API objets

Du point de vue du développeur, les accès sont aussi possibles de plusieurs façons. Le métamodèle de données et l'entrepôt sont persistants dans un SGDB relationnel. Les données sont donc directement interrogeables en SQL. Le modèle étant extensible, il est aisément possible de créer des vues et des procédures stockées orientées métier. Nous proposons de plus une API en Java orientée objets et pour manipuler les données de l'entrepôt. Il est bien sûr possible de proposer des API dans d'autres langages disposant de connecteurs vers le serveur choisi (MySQL dans notre cas). Enfin, nous avons mentionné la possibilité d'utiliser les standards des services Web.

Cette souplesse offerte par le métamodèle présente cependant plusieurs limites. Dans un premier temps, ces limites sont liées aux performances. Concernant l'extensibilité du modèle, nous avons pallié le problème en proposant deux catégories de propriétés : les propriétés statiques (type, source, preuve, organisme) et les champs (cf. section 4.2.2). Cette solution s'avère efficace. Dans un second temps, le métamodèle rassemble dans les relations des nœuds, des champs et des arêtes un très grand nombre de tuples. Un modèle du domaine aurait produit des relations plus nombreuses possédant de plus nombreux attributs, mais un nombre bien plus faibles de tuples. Ce grand nombre de tuples représente un coût dans différentes manipulations des entités, malgré les optimisations par des clés et des index.

Un dernier problème se pose en matière d'expressivité du schéma : il est possible d'exprimer de nombreuses relations entre les éléments, mais la structuration du schéma à l'aide de contraintes d'intégrités s'avère très coûteuses. Elle nécessite de construire des contraintes à partir de procédures stockées complexes, associées à des déclencheurs (*triggers*).

Le métamodèle de graphe : une approche intuitive

Dans une base de données, la notion de jointure est complexe. Issue d'un algèbre, elle relève d'une abstraction mathématique rarement maîtrisée par l'utilisateur final. Lorsque ce dernier définit un chemin, le mécanisme est comparable à une jointure naturelle. Les expériences menées par les systèmes à base de chemins comme Getz montrent que ce mécanisme est mieux compris et manipulé par les non-informaticiens. Cet aspect intuitif provient entre autre de ce qu'un graphe est une structure simple en comparaison du modèle relationnel qui s'appuie sur une théorie ensembliste. Notre métamodèle structure les données en graphe de façon à ce que l'utilisateur final ait une meilleure compréhension du système qu'il manipule. Comme nous l'avons vu, cela n'entame pas l'expressivité du modèle du point de vue du développeur. La simplification du modèle permet au développeur, entre autres, de cerner plus rapidement les contours du système d'information et de concevoir ou mettre à jour plus rapidement des procédures d'intégration ou de fouille de données.

Une boîte à outils graphique simple à mettre en œuvre

La boîte à outils graphique que nous proposons fournit un grand nombre de fonctionnalités permettant de construire des clients riches rapidement. Elle étend une boîte à outils existante, Prefuse, qui a déjà été évaluée. Les résultats sont très convaincants : en moins d'une heure, le programmeur peut appréhender la boîte à outils et commencer à obtenir un résultat élémentaire. Les exemples de code que nous avons fournis dans les deux chapitres précédents ont montré la simplicité de l'ajout de fonctionnalités. Ce code est assimilable à du script : il est déclaratif et non algorithmique.

On peut, par ailleurs, très facilement envisager de déporter une partie de ce code dans des fichiers XML. Ceci est déjà partiellement réalisé pour les menus contextuels et peut être très simplement mis en œuvre pour les lentilles et les sélections en utilisant les mécanismes d'injection de dépendance de Spring par exemple.

8.2.2 Aspects fonctionnels et interactions utilisateurs

L'utilisateur et le développeur ont des besoins hétérogènes et distincts. La première partie de cette synthèse a adressé essentiellement les aspects techniques liés au développement. Dans cette seconde partie, nous proposons de nous intéresser plus spécifiquement aux problèmes de l'utilisateur final et souhaitons montrer comment I²DEE répond à l'essentiel des besoins. Les difficultés de l'utilisateur, rappelons le, consistent essentiellement à :

- lier les données expérimentales aux connaissances partagées du domaine et favoriser l'analyse de ces données,
- comparer les données issues de différentes sources,
- favoriser l'échange de données (interopérabilité entre applications, export graphique ou vers des tableurs, etc.).

Globalement, notre approche pour répondre à cela ne consiste pas à proposer une plateforme unique contenant des fonctionnalités pour toutes les tâches métier du biologiste. I²DEE vise à proposer un cadre commun souple, extensible, adaptable, pour créer des portails, des outils métiers et les rendre interopérables et homogènes.

Dans le chapitre 3, nous avons énuméré plusieurs objectifs concrets à poursuivre afin d'améliorer les conditions d'accès, de manipulation et d'analyse de l'information. Différentes propositions en découlent dans les chapitres suivant concernant l'architecture et la boîte à outils graphique. La suite de cette section synthétise et détaille chacune de ces propositions.

Réduire le nombre de fenêtres

Nous avons dans un premier temps résumé le problème d'utilisateur au nombre de fenêtre manipulées. Nous avons par la suite évoqué la multiplicité des sources, leur hétérogénéité et la difficulté de comparer ou de recouper les informations concernant un élément donné (gène, etc.).

I²DEE, de par sont modèle d'intégration, son architecture ou encore la visualisation qu'il propose répond à ces différents besoins. Tout d'abord, l'accès à l'information est centré sur l'élément de données. Par exemple, l'utilisateur sélectionne un gène et accède directement à son voisinage, c'est-à-dire à ses annotations, ses transcrits, la bibliographie qui lui est relative, etc. L'accès à cette information se fait directement suivant différentes sources. On évite ainsi la multiplication des pages concernant un même gène partagé par différentes sources. Notons par ailleurs que l'utilisateur n'est pas non plus obligé de connaître au préalable les sources qui lui sont nécessaires.

De plus, si l'utilisateur souhaite étudier plusieurs gènes, la visualisation permet de consulter l'information dans une même fenêtre. Le recoupement des données se fait alors visuellement comme l'ont montré les captures du chapitre précédent : si plusieurs gènes ont une même annotation, ils sont reliés au même concept et sont rapprochés. Il n'est plus nécessaire de lire séparément les annotations de chaque gène dans des pages Web différentes (et donc des fenêtres différentes) pour identifier les éléments communs. On allège ainsi la charge cognitive engendrée. Nous conservons à nouveau l'information relative à plusieurs gènes au sein de la même fenêtre et réduisons ainsi le nombre de fenêtres manipulées par l'utilisateur.

Enfin, le plus souvent, l'utilisateur accède à ces informations au travers de portails en ligne alors qu'il manipule les éléments biologiques (gènes, protéines, etc.) dans une application métier. En intégrant la visualisation des données analysées et des connaissances partagées, l'utilisateur n'utilise plus qu'une seule application.

Si nous poursuivons l'objectif que l'utilisateur ne manipule qu'un nombre réduit d'outils basés sur I²DEE pour accéder à l'information, il souhaite parfois vérifier le contenu ou le compléter en accédant à un portail, ou encore à des services de ce portail. I²DEE ne prétend pas supprimer totalement ce besoin mais simplement le réduire. Au contraire, il en facilite la

démarche : un menu contextuel facilement personnalisable permet d'ouvrir simplement la page de PubMed relative à un document. Cependant, en proposant auparavant le résumé par un survol de la souris, l'utilisateur est amené à lancer le navigateur vers PubMed uniquement si le document l'intéresse.

Gestion de la surcharge d'information

Au travers d'un portail, toutes les informations sont présentées concernant un seul gène ou une seule protéine par exemple. L'intégration de l'information relative à plusieurs gènes et plusieurs sources, au sein d'une seule fenêtre engendre potentiellement une surcharge d'information. Pour pallier cette limite, nous proposons plusieurs mécanismes :

- les applications dédiées à des tâches particulières permettent de limiter les types de données nécessaires,
- les lentilles permettent d'afficher l'information qui concerne uniquement les éléments de données sur lesquels se focalise l'utilisateur,
- la visualisation permet de faire émerger des relations directes ou indirectes communes à plusieurs éléments (effet de regroupement),
- la boîte à outils fournit des mécanismes multiéchelles (zoom, aperçu, etc.).

Notons que certains logiciels permettent de traiter globalement l'information massive. Par exemple, certains outils proposent une annotation probable concernant un ensemble de gènes. Cependant, ces systèmes sont opaques : ils mettent en œuvre des algorithmes complexes produisant une valeur de score final et masquent toute la trace du calcul. I^2DEE au contraire fait émerger visuellement l'information, mais ne la masque pas pour autant. Si des annotations majoritaires concernant un ensemble de gènes sont rapidement perçues par l'utilisateur, les informations alternatives seront aussi mises en évidence. Nous avons vu en comparant deux regroupements automatiques qu'il ne s'agit pas d'obtenir un regroupement remplaçant les deux précédents. Au contraire, nous mettons en valeur leurs convergences et leurs divergences qui peuvent diriger l'utilisateur dans le choix d'affiner certains paramètres, d'isoler plusieurs gènes d'intérêt, etc.

Des outils multiples mais homogènes

Au travers d'I^2DEE, nous souhaitons homogénéiser l'interface utilisateur. Pour cela, notre boîte à outils propose une visualisation commune aux différentes tâches et nous souhaitons limiter l'usage des portails. Pour autant, nous proposons d'utiliser de multiples applications basées sur I^2DEE pour l'accès aux données intégrées et leur visualisation. On résout ainsi de nombreux problèmes d'hétérogénéité grâce à différents mécanismes :

- la topologie des données est commune aux différentes applications (un graphe),
- la méthode de visualisation l'est aussi,
- les menus contextuels et plus généralement les interactions sont gérés de façon commune,
- une feuille de style propre à l'utilisateur lui permet de définir l'ordre dans lequel il souhaite présenter l'information, les couleurs correspondant aux données, ou encore les données qu'il souhaite filtrer par exemple.

Echange de données – Interopérabilité

Les différentes applications partagent le même modèle de données et par conséquent peuvent partager serveurs et fichiers. Cela évite tous les problèmes courants que rencontre l'utilisateur dans l'import ou l'export de fichier, la difficulté de configurer des serveurs et s'inscrire pour chaque nouvelle application. Cela permet aussi de rendre commun le modèle du « presse-papier » et donc de faciliter les « copier/coller » entre applications. Les applications, en étant centrées sur des données dont le modèle est unifié sont directement interopérables.

De la même façon, une sélection multiple est un sous graphe, et peut facilement être rendue persistante au sein d'un fichier, ou transférée vers une autre application ou un autre fichier. Toutes ces considérations concernant l'interopérabilité sont soumises à la condition que les différentes données partagées et distribuées soient synchronisées dans la gestion des identifiants.

Une approche extensible et adaptable pour des outils conviviaux et intuitifs

Pour qu'I²DEE puisse être adopté par une large communauté d'utilisateurs, nous l'avons architecturé et conçu d'une façon ouverte : il n'y a pas de restriction majeure sur la nature des données, la source, ou le mode d'accès aux données. De même, la boîte à outils est conçue pour répondre aux besoins les plus divers :

- la technique de visualisation est robuste, polyvalente et peut s'adapter dynamiquement à différentes topologies de graphe,
- d'autres techniques de visualisation complémentaires sont adjointes et synchronisées au travers des mécanismes de sélection,
- elle dispose d'outils génériques permettant de spécifier des fonctionnalités métier : la gestion des types peut être décorrélée entre l'entrepôt et la visualisation, la définition d'une lentille est décomposée en objets génériques (modifieurs, chemins, filtres, évènements d'interaction), outils de recherche textuel peuvent être associés à tous les types et à tous les attributs de données, etc. ,
- le mécanisme de requête semblable à ceux des systèmes à base de liens permet de spécifier des requêtes possédant des jointures de façon naturelle pour l'utilisateur.

Toutes ces adaptations sont réalisables dans une syntaxe plus proche du script que de la programmation et peuvent être externalisés facilement en XML via les mécanismes d'injection de dépendance de Spring par exemple.

8.3 Discussions : des résultats aux perspectives

Nous avons présenté dans la seconde partie de ce mémoire de nombreuses perspectives. Nous n'en reprenons pas ici un inventaire exhaustif, mais les synthétisons dans quelques directions scientifiques qui nous paraissent importantes. Nous n'abordons pas les questions techniques et ponctuelles comme l'amélioration des performances durant l'intégration, l'analyse textuelle, ou la visualisation. Nous ne discutons pas non plus des aspects légaux liés à l'exploitation de données issues de multiples sources. La suite de cette section s'intéresse aux points suivants :

- l'évaluation des résultats,
- les évolutions fonctionnelles visant à proposer des mécanismes de haut niveau pour adapter plus facilement l'outil aux différents domaines, utilisateurs et usages,
- l'intégration non seulement de données mais aussi de procédure, ce qui nécessite la prise en charge de paramétrage, de génération de données à la volée, et la gestion de données persistantes temporaires et intermédiaires.

8.3.1 Evaluation des résultats

L'évaluation des différentes techniques mises en œuvre est déterminante, mais s'avère particulièrement difficile. Nos travaux sont étendus sur plusieurs domaines interdépendants : architecture logicielle, intégration de données, fouille de textes et indexation documentaire, analyse de données par regroupements automatiques, ingénierie des connaissances, visualisation d'information, interactions homme-machine et adaptabilité de l'interface. Il est nécessaire de proposer des critères objectifs et quantitatifs afin d'évaluer de façon indépendante

chacune des techniques mise en œuvre depuis la construction de l'entrepôt jusqu'à l'utilisation finale afin de valider et d'optimiser chaque étape.

Le domaine d'application

La première difficulté est inhérente au domaine d'application : la biologie. Contrairement à d'autres domaines de recherche, nous ne possédons pas de jeu d'essai nous permettant de comparer nos résultats. Les résultats des expériences des biologistes sont variables, difficilement comparables à d'autres expériences. Les connaissances concernant l'organisme *Plasmodium Falciparum* sont par ailleurs peu nombreuses : la plupart des protéines et des annotations associées à des gènes ont été obtenues par des algorithmes et n'ont pas été validées individuellement par un expert. Il est nécessaire de construire dans un premier temps un jeu d'essai fiable concernant un organisme pour lequel les recherches sont plus avancées. Le projet concernant la leucémie nous permettrait de mener une évaluation plus précise. Le génome humain est connu depuis plus longtemps que celui de Plasmodium. Les ressources sont plus nombreuses : OMIM partage des synthèses bibliographiques, HUGO unifie la nomenclature des gènes, etc. Les annotations validées individuellement par des expérimentations et annotées par un expert sont plus fréquentes. De plus, le chercheur est fortement spécialisé : il connait et explore des domaines précis. Le jeu d'essai doit concerner son champ de recherche et ses besoins précis courants. Il parait difficilement envisageable de mener des évaluations sur de multiples utilisateurs comme cela peut être réalisé dans d'autres contextes : chaque expérimentation nécessite de posséder (ou produire) un outil adapté à la tâche, d'intégrer les bases de données du domaine, de construire un jeu d'essai à partir des données expérimentales, et de proposer des alternatives avec lesquelles comparer l'outil.

Interconnexions entre différentes domaines informatiques

La seconde grande difficulté résulte de l'étendue de nos recherches sur deux grands domaines informatiques : le traitement des données, et leur manipulation interactive par l'utilisateur. Il est possible de comparer des méthodes de traitement de données à l'aide de différentes métriques. De la même façon, certaines évaluations d'interfaces utilisateurs sont réalisables à partir d'un jeu de données considéré comme satisfaisant par l'utilisateur final. L'évaluation des méthodes d'intégration n'est pas déterminante. Notre contribution n'est pas de proposer de nouvelles méthodes de fouille, et si l'évaluation de ces méthodes se révèle difficile, il est encore plus difficile d'en quantifier l'impact sur la globalité d'I^2DEE. Cet impact n'est pourtant pas négligeable : lorsque l'utilisateur est placé devant l'outil, il lui est difficile de considérer l'évaluation fonctionnelle de l'outil en faisant abstraction de la qualité du contenu. La satisfaction de l'utilisateur dépend de ce que les informations lui apprendront. Elle dépend donc : de la technique d'analyse de données, des méthodes d'intégration et de fouille de données, des sources de données sélectionnées, et plus généralement du domaine. Dans le contexte de *Plasmodium Falciparum*, l'utilisateur nous indique que l'outil est intéressant, mais il émet des réserves sur l'apport global de l'expérience du fait du manque de connaissances de la communauté sur cet organisme.

Evaluation globale et qualitative

Cependant, devant la difficulté de décomposer I^2DEE en composantes évaluées indépendamment et de mener de telles expérimentations dans les délais impartis, nous avons approché l'évaluation différemment, à un niveau global. Nous proposons d'utiliser les critères suggérés dans le chapitre 3 pour évaluer qualitativement la pertinence de notre approche. La première partie de ce chapitre a synthétisé ces critères en une question simple : I^2DEE permet-il de réduire le nombre de fenêtres manipulées par l'utilisateur, et de simplifier le travail de comparaison et de synthèse des résultats ? Si nous ne pouvons répondre à ces questions par un indice quantitatif, nous pouvons apporter dans la suite quelques éléments de réponse. A la première partie de la question, concernant le nombre de fenêtres, proposons les constats suivants :

- I²DEE ne génère pas plus de fenêtres qu'auparavant (par « l'absurde »),
- une fenêtre d'I²DEE remplace une ou plusieurs applications métiers,
- I²DEE permet de consulter les informations relatives à plusieurs éléments (gènes, protéines) et provenant de plusieurs sources dans la même fenêtre.

Généralement, *n* gènes et *m* sources produisent de l'ordre de $n \times m$ fenêtres de navigateur. I²DEE permet d'envisager de n'en avoir plus qu'une. S'il n'y a aucune garantie d'obtenir ce résultat optimal, on peut espérer réduire drastiquement le nombre de fenêtres.

La seconde partie de notre question concerne le croisement de l'information provenant de plusieurs sources avec les données expérimentales. A nouveau, on peut réaliser plusieurs constats qualitatifs : les données possédant des points communs seront reliées directement à des nœuds communs, et produisent donc un effet similaire à celui d'un regroupement automatique. Le croisement des données est ainsi facilité :

- entre plusieurs éléments (gènes, protéines, etc.),
- entre plusieurs sources,
- entre les connaissances partagées du domaine et les données expérimentales qui sont au cœur de la visualisation.

Grâce à I²DEE, l'utilisateur ne doit plus systématiquement recourir à des supports externes pour la cognition tels qu'une feuille de papier ou un tableur dans lequel il copie et colle de multiples informations. La mémoire de l'utilisateur est elle-même sollicitée au minimum.

Le dernier point sur lequel une évaluation quantitative est importante est celui de l'adaptabilité. Nous avons proposé un mécanisme simple reposant sur des algorithmes d'analyse de liens robustes, et dont les comportements et les aléas sont bien identifiés dans la littérature [Borodin, Roberts et al. 2005]. Il est cependant nécessaire d'évaluer les bénéfices de ces algorithmes, leurs effets de bord et les optimisations nécessaires. Une telle évaluation doit être quantitative et s'appuyer sur un jeu d'essais. L'intérêt émerge d'une l'utilisation prolongée, éventuellement par une équipe et concernant plusieurs tâches. Comment évaluer la qualité du résultat qui ne peut objectivement être comparé différentiellement avec une utilisation sans adaptabilité. Si cette fonctionnalité semble intéressante *a priori*, une nouvelle démarche d'évaluation doit être construite pour ce problème, car les mécanismes d'adaptabilité qui ont été étudiés jusqu'ici concernaient des utilisations à court termes sur des produits grand public et non sur des applications très spécialisées.

8.3.2 Mécanismes d'adaptation de haut niveau

Nous avons montré la proximité entre la programmation de certaines fonctionnalités, et les scripts qui pourraient être générés. De façon directe et sans effort, il est possible d'injecter des objets Java au travers de la bibliothèque Swing. Cela offre la possibilité de télécharger et mettre à jour des fichiers de configurations sans avoir à recompiler l'application. Cela permet de proposer facilement plusieurs adaptations, *Look & Feel*[1], à l'utilisateur, ou d'introduire des *plug-ins*. Mais ces solutions restent des outils pratiques accessibles au développeur, toujours trop complexes pour l'utilisateur final.

Une particularité de notre approche de l'intégration réside dans la présence d'un métamodèle de graphe. Il offre une grande souplesse et s'avère central dans de nombreux aspects. La principale direction de nos recherches consiste à s'appuyer sur ces métadonnées pour proposer des outils d'adaptation par script génériques, complets, et homogènes. Ces scripts s'appliqueraient à :

- des feuilles de style pour les clients riches et les portails en ligne génériques,

[1] Charte graphique décrivant l'apparence visuelle d'une interface et les interactions associées.

- des spécifications de requêtes pour copier les données dans des tableurs, intégrer des sources, interroger les données depuis des systèmes à base de lien, spécifier des services Web, la définition de méthode d'extraction de carte, la définition de méthodes de propagation de valeurs de pondération (pour l'adaptabilité notamment), etc.
- des spécifications fonctionnalités avancées pour les clients riches comme les lentilles métier, les fonctions de recherche, etc.

Feuille de Style

La feuille de style que nous proposons permet d'ordonner certaines métadonnées et de leur attribuer des propriétés. Le principe est de proposer une charte graphique personnalisée partagée par les différentes applications manipulées par l'utilisateur. Cette charte graphique permet de définir :

- les données qui seront prises en compte par le client (il est possible par exemple de filtrer des types de nœuds ou de relations précis),
- l'ordre dans lequel ces données doivent être présentées,
- leurs propriétés graphiques relatives (couleur de fond, de texte, taille de la police, visibilité par défaut, transparence, rayon, etc.).

Cette charte est utile dans deux contextes : l'utilisation d'un client graphique lourd développé avec notre boîte à outils et l'utilisation d'un portail en ligne avec une interface hypertextuelle (HTML). Le choix de filtrer les données permet d'éviter l'affichage d'information inutile. L'utilisateur peut par exemple décider de ne pas afficher les séquences qui sont généralement volumineuses à l'écran et dans les transferts de données. Ces fonctionnalités sont cependant déjà présentes dans le « contrôleur » de notre boîte graphique. Dans le contexte d'un portail textuel, le graphe n'étant pas dessiné, les relations n'apparaissent pas. Elles correspondent au contenu qui est affiché : sélectionner une relation d'annotation entre un gène et un concept implique qu'on affiche pour un gène donné ses annotations. De la même façon, on peut définir qu'on souhaite afficher les références bibliographiques, les *locus*, les protéines traduites, etc. L'ordre est lui aussi important : on peut afficher en tête de document des informations fonctionnelles et déporter à la fin la séquence comme cela est couramment pratiqué dans la plupart de portails existants. Cet ordre est d'autant plus important lorsqu'il s'agit de type de relations dans le contexte d'un portail : il est possible de spécifier qu'on souhaite afficher les annotations en premier, les protéines traduites ensuite, et les références bibliographique en spécifiant l'ordre des relations relatives à un gène. Enfin, l'utilité des propriétés graphiques se déduit facilement (couleurs, police, tailles, transparence, visibilité, etc.).

Dans le contexte de clients graphiques lourds basés sur notre boîte à outils, la mise en œuvre de ces feuilles de style est assez simple. Les propriétés graphiques sont déjà centralisées dans les types graphiques. L'injection de dépendance permet de déporter cela facilement vers du XML. Les attributs peuvent être affichés dans un panneau générique associant dans une première colonne les noms des champs à leurs valeurs inscrites dans la seconde colonne. L'ordre d'affichage est aussi simple à prendre en compte. Reste la notion de sélection des types à afficher. A nouveau, l'architecture de la boîte à outils permet d'intégrer cette fonctionnalité dans les filtres. Mais comment doit-on gérer le conflit suivant : un utilisateur filtre un type de donnée nécessaire à l'application ? L'application n'est pas obligée de prendre en compte la charte graphique ; il s'agit d'une fonctionnalité qu'elle peut intégrer ou non.

Concernant le moteur de services de pages HTML, aucune implémentation n'a été entamée. La particularité de la page de texte est de ne pas afficher le graphe. Par conséquent, les relations sont nécessaires pour indiquer les informations relatives à un élément donné à afficher. Par exemple, la sélection d'une relation d'annotation, d'une relation de traduction vers une protéine et d'une relation de référence bibliographique implique que concernant un gène donné, on affichera respectivement ses annotations suivies des protéines qu'il code et des références bibliographiques. Des chemins de longueur deux doivent alors être présents pour indiquer qu'on souhaite connaître les auteurs pour chaque document. Si nous n'avons entrevu actuellement

aucune limite dans la faisabilité de ce serveur, des problèmes risquent d'apparaître dans la gestion des performances.

Par ailleurs, trois limites apparaissent d'un point de vue plus fonctionnel. La première est de savoir comment croiser les informations : même si on peut afficher plusieurs gènes de façon homogène dans une même page, comment faciliter le recoupement d'informations ? La notion d'agrégation des données telle qu'elle existe dans les bases de données est-elle suffisante ?

La seconde limite est plus globale à l'ensemble du système : les informations présentes dans cette feuille de style sont nombreuses. Elles sont suffisamment complètes pour spécifier le comportement d'un portail complet, et pour spécifier plus que l'apparence d'un client graphique. Cela dépasse le simple *« look & feel »* et s'adresse au développeur. Ces feuilles de style deviennent trop complexes pour l'utilisateur. Nous proposons d'envisager l'existence de deux types de feuilles : une feuille complexe pour le système, et une feuille plus simple pour l'utilisateur. Pour faire interopérer les deux, on introduit alors la notion de contexte parent, telle qu'il existe dans Spring, et qui permet d'hériter ou de redéfinir certaines propriétés.

Enfin, la troisième limite provient de l'aspect statique de ces données : lorsque nous initialisons notre vue, par exemple, le rayon des regroupements automatiques est calculé de façon non linéaire à partir du nombre de nœuds lui étant associés. La question est de savoir comment permettre à ces fonctions de spécifier de telles opérations, et comment accéder aux différentes propriétés. Cela nécessite des recherches plus profondes que les deux limites précédentes, d'autant que ce calcul doit être réalisé uniquement à l'initialisation de l'outil pour assurer des performances satisfaisantes.

Requêtes : sélections, projections, jointures

Considérons que chaque type de données de notre entrepôt donne lieu à une relation (ou table). Chaque tuple (ligne) correspond à un gène par exemple, chaque colonne à un attribut du gène (nom, séquence, etc.). On peut assimiler les opérations précédentes à l'algèbre relationnelle :

- on choisit les relations présentes dans une requête,
- on choisit les attributs que l'on conserve (projection),
- on choisit les arêtes que l'on conserve (jointures).

Les opérations réalisées par le navigateur sont donc proche de l'algèbre relationnelle. Elles sont plus simples et intuitives pour l'utilisateur, mais leur expressivité est limitée. Cependant, si on considère que le résultat est tabulaire, on peut immédiatement utiliser une simple requête de type (gène – *traduction* – protéine – *annotation* - concept) pour produire un tableau contenant pour chaque gène les protéines traduites et les concepts annotant ces protéines. Dès lors que ce tableau est conservé dans un format standard, il peut être exporté vers le tableur favori du biologiste.

Le tableur est un outil particulièrement précieux du chercheur. Une direction que nous souhaitons suivre est donc d'étudier en profondeur les mécanismes que l'on peut exprimer, appliquer du coté de l'entrepôt pour générer un document et exploiter efficacement dans un tableur. A termes, la question se pose aussi de conserver la traçabilité nécessaire à la réimportation des données du tableur dans l'entrepôt de données. Une autre problématique concerne l'introduction d'un mécanisme équivalent à la sélection de l'algèbre relationnelle : par exemple, ne conserver que les documents de moins de 5 ans, les annotations manuelles, etc.

Les multiples intérêts des scripts sont maintenant plus clairs et concrets. Leur mise en œuvre ne concerne pas uniquement le client lourd et le portail HTML généraliste. Les opérations décrites constituent un véritable langage de requêtes simplifié. Il est donc pertinent d'envisager l'extension de ces méthodes de scripts :

- dans les étapes d'intégration en s'inspirant du langage Getz ou d'autres initiatives de la communauté pour spécifier des adaptateurs,
- pour spécifier des méthodes d'extraction de cartes contextuelles,

- pour la définition de méthodes de propagation de pondération dans les fonctionnalités d'adaptabilité du système,
- pour générer des vues métier de l'entrepôt,
- pour générer des services Web,
- etc.

Fonctionnalités métier du client

Nous avons montré qu'il est possible de définir en Java des lentilles métier ou encore des fonctions de recherche. Le code procédural correspondant est linéaire, simple et peut être aisément externalisé dans du code XML via un mécanisme d'injection de dépendance, par exemple. Mais là encore, les scripts utiliseraient des valeurs statiques : comment mettre en œuvre des mécanismes permettant de spécifier des calculs plus fonctionnels, par exemple pour définir la longueur d'une arête ou encore le rayon d'un regroupement ? Par ailleurs, des formalismes sont proposé afin de spécifier les vues qui peuvent nous inspirer dans [Tricot 2006].

Une spécification de la vue métier à un plus haut niveau

Toutes les fonctionnalités que nous envisageons nous semblent réalisables et particulièrement intéressantes et pertinentes concernant le problème concret qui nous a été présenté en début de thèse. Mais un défi soulevé par Crampes et al. nous semble plus audacieux et nous motive à plus long termes [Crampes, Villerd et al. 2006]. Il s'agit de permettre une description de la vue suivant une conceptualisation de haut niveau. Cette conceptualisation permettrait alors de générer le code nécessaire (en Java ou en scripts XML) et permettraient de limiter le besoin pour le développeur comme pour l'utilisateur de s'investir dans le paramétrage de l'application et en bénéficiant de capacité de spécialisation pour une tâche métier, de personnalisation et d'adaptation.

8.3.3 Des services

> *With a connection to the Internet and a web browser it is possible to access easily almost all existing data banks and software devoted to sequence study. Some limitations still exist in the use of such servers. A significant problem is that it is often not possible to keep track of the previous interrogations from one session to another: once the user is disconnected from the server, all his/her work is lot. [Perrière, Combet et al. 2003]*

Une dernière problématique qui nous intéresse concerne les services. Elle est mise en évidence notamment par Guy Perrière dans [Perrière, Combet et al. 2003]. Il montre dans cet article qu'une lacune de nombreux systèmes en lignes est qu'ils partagent des informations et proposent des services, mais ne permettent pas de les enchaîner. Ils ne possèdent en effet aucun mécanisme de persistance des données intermédiaires. Seuls les systèmes de flux de travail permettent d'enchaîner des traitements de telle sorte.

L'ouverture d'I^{2}DEE concernant cette problématique est la suivante : il est possible d'accéder à l'information sous forme de services. Les traitements sont d'autres services (présents au sein d'I^{2}DEE ou non) combinés à l'aide de clients comme Taverna par exemple.

Nos propositions dans ce cadre consistent à modéliser les services comme des procédures dont les paramètres sont liés aux métadonnées de l'entrepôt. Chaque procédure devient alors un nœud d'I^{2}DEE et génère des données à la volée. Une donnée intermédiaire est une sélection persistante. Un problème est alors de représenter des relations n-aires.

De plus, nous réfléchissons à considérer des visualisations complémentaires comme des services paramétrés par une sélection de données et une feuille de style. On peut ainsi facilement ajouter de nouvelles fonctions, découplées de l'architecture d'I^{2}DEE. L'introduction

d'une nouvelle visualisation est alors plus simple et adaptée à des besoins spécifiques comme l'analyse de séquence ou l'alignement multiple par exemple [Duret, Gasteiger et al. 1996].

8.4 Conclusion

Les travaux présentés dans ce mémoire concernent l'intégration et la visualisation de données. L'objectif était d'offrir l'accès à différents types de ressources *via* une carte des connaissances. Nous l'avons appliqué à la problématique de l'ingénierie des connaissances et de l'analyse de données d'expression de gènes en sciences du vivant. Un environnement a été développé : I^2DEE.

L'intégration de données hétérogènes est réalisée au travers d'un modèle de graphe qui constitue un métamodèle. Le niveau *meta* procure souplesse et extensibilité. La capacité d'introspection qu'il propose permet d'adjoindre un outillage générique et paramétrable. Ce modèle permet l'ajout de ressources et une utilisation de l'entrepôt simplifiés. Quelques limites existent en termes de performances et d'expression de contraintes sur le schéma.

De cet entrepôt de données cartographiques du vivant, nous réalisons une extraction contextuelle de carte. La carte est alors un support dont l'utilisateur est propriétaire. Elle est visuelle et s'inspire d'algorithmes d'analyse de liens. La carte est visualisée dans une boîte à outils dont l'objectif est de permettre la conception facile de multiples applications interopérables, visuelles, et adaptées à des tâches spécifiques. Au travers de ces interfaces graphiques, le biologiste manipule ses données expérimentales et les croise avec les connaissances partagées du domaine. Il n'est plus contraint de multiplier les pages Web sur son poste de travail et s'économise de nombreux copier/coller voir l'usage d'une structure intermédiaire pour synthétiser l'information (généralement un tableau).

Les perspectives sont nombreuses. Elles consistent dans un premier temps à mettre en œuvre l'environnement dans un jeu d'expérience complet à dimension réel et de procéder à des évaluations plus profondes de notre contribution. Dans un second temps, nous souhaitons diriger nos réflexions vers la spécification à un plus haut niveau des composantes de l'environnement : requêtes, adaptateurs, procédures d'extraction et d'adaptabilité, feuilles de style, définition de l'application, etc.

En définitive, notre prochain travail est celui du cartographe : il consiste à étudier les méthodologies pour construire les cartes de connaissances biologiques et à proposer un outillage adapté.

> *« [Ce livre] aborde une problématique à laquelle sont confrontés quotidiennement beaucoup de professionnels : celle de comprendre et de gérer leurs territoires grâce à la cartographie. L'enjeu est grand : meilleurs seront les documents cartographiques, plus sûres seront les décisions qui en émanent. »*
>
> ***DIDIER POIDEVIN***

Références bibliographiques

The Gene Ontology.

Gene Ontology Annotation.

PubGene Gene Database and Tools.

Abramowitz, M. and I. A. Stegun (1972). Handbook of Mathematical Functions with Formulas Graphs, and Mathematical Tables. New York, Dover Publications.

ABU. "ABU : la bibliothèque universelle." from http://abu.cnam.fr/DICO/.

Achard, F., G. Vaysseix, et al. (2001). "XML, bioinformatics and data integration." **17**(2): 115-125.

Allen, J. F. (1984). "Towards a general theory of action and time." Artificial Intelligence in Medicine **23**: 123-154.

Andrieu, D. (2005). "L'intérêt de l'usage des cartogrammes: l'exemple de la cartographie de l'élection présidentielle française de 2002." Mappemonde **77**.

Anoir, I., J.-M. Penalva, et al. (2005). L'interaction dans les collectifs collaboratifs ; typologie et perspectives. Enjeux et Usages des TIC - Aspects Sociaux Culturels, Université de Bordeau 3, France.

Apweiler, R., A. Bairoch, et al. (2004). "UniProt: the Universal Protein knowledgebase." Nucleic Acids Research **32**(Database-Issue): 115-119.

Arens, Y. and C. A. Knoblock (1993). SIMS: Retrieving and Integrating Information From Multiple Sources. Proceedings of the 1993 ACM SIGMOD (Special Interest Group on Management of Data) International Conference on Management of Data, Washington, D.C, Washington, D.C., ACM Press.

Arnheim, R. (1969). Visual Thinking. Berkeley (California, USA), University of California Press.

Arpírez, J. C., O. Corcho, et al. (2001). Webode: a scalable workbench for ontological engineering. First International Conference on Knowledge Capture (K-CAP 2001).

Arrivé, M. (2006). L'Orthographe pour tous, Hatier.

Ashburner, M., C. A. Ball, et al. (2000). "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium." Nature Genetics **25**(1): 25-29.

Auber, D., Y. Chiricota, et al. (2003). Multiscale Visualization of Small World Networks. IEEE Symposium on Information Visualization (InfoVis 2003).

Auber, D., M. Delest, et al. (2004.). Strahler based graph clustering using convolution. 8th International Conference on Information Visualisation, EEE Computer Society**:** 44-51.

Aurrecoechea, C., M. Heiges, et al. (2007). "ApiDB: integrated resources for the apicomplexan bioinformatics resource center." Nucleic Acids Research **35**(Database issue): 427-430.

Aussenac-Gilles, N. and D. Bourigault (2003). Construction d'ontologies à partir de textes. TALN'03 - Traitement Automatique du Langage Naturel.

Baader, F., D. Calvanese, et al. (2003). The Description Logic Handbook - Theory, Implementation and Applications, Cambridge University Press.

Bachimont, B. (2000). Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en Ingénierie des connaissances. J. Charlet, M. Zacklad, G. Kassel and D. Bourigault, Ingénierie des connaissances, évolutions récentes et nouveaux défis - Eyrolles.

Baduel, L., F. Baude, et al. (2006). Programming, Deploying, Composing, for the Grid. Grid Computing: Software Environments and Tools. J. C. Cunha and O. F. Rana, Springer-Verlag.

Bahl, A., B. Brunk, et al. "PlasmoDB." NAR Molecular Biology Database Collection, entry number 91.

Baker, P. G., A. Brass, et al. (1998). TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources. 6th International Conference on Intelligent Systems for Molecular Biology(ISMB98), Montréal, Canada, AAAI Press.

Ball, C. A. and A. Brazma (2006). "MGED Standards: Work in Progress." OMICS: A Journal of Integrative Biology **10**(2): 138-144.

Ball, C. A., A. Brazma, et al. (2004). "Submission of Microarray Data to Public Repositories." PLoS Biology **2**(9).

Baril, X. (2003). Un modèle de vues pour l'intégration de sources de données XML : VIMIX Informatique. Montpellier, Université Montpellier II. **Ph.D**.

Barkowsky, T. and C. Freksa (1997). Cognitive Requirements on Making and Interpreting Maps. Spatial information theory: A theoretical basis for GIS. S. C. Hirtle and A. U. Frank. Berlin, Springer**:** 347-361.

Barnes, J. and P. Hut (1986). "A hierarchical O(N log N) force-calculation algorithm." Nature **324**: 446-449.

Baru, C., A. Gupta, et al. (1999). "XML-Based Information Mediation with MIX." SIGMOD '99 (demonstration session).

Bass, L., R. Faneuf, et al. (1992). "The UIMS Workshop Tool Developers : A Metamodel for the Runtime Architecture of an Interactive System." SIGCHI Bulletin **24**(1): 32-37.

Baumbach, J., K. Brinkrolf, et al. (2006). "CoryneRegNet: An ontology-based data warehouse of corynebacterial transcription factors and regulatory networks." BMC Genomics **7**(24).

Bechhofer, S., I. Horrocks, et al. (2001). OilEd: a reasonable ontology editor for the semantic web. Joint German/Austrian conference on Artificial Intelligence, KI2001, Springer-Verlag.

Bederson, B. B. (2000). Fisheye Menus. ACM Conference on User Interface Software and Technology (UIST 2000), ACM Press.

Benedikt, M. L. (1991). Cyberspace: Some proposals. Cyberspace: First steps. M. L. Benedikt. Cambridge (Massashusetts), MIT Press.

Beneventano, D., S. Bergamaschi, et al. (2000). Information Integration: the MOMIS Project Demonstration. Proceedings of 26th International Conference on Very Large Data Bases (VLDB 2000), Le Caire (Egypt), Morgan Kaufmann Publishers.

Bennouas, T. (2005). Modélisation de Parcours du Web et Calcul de Communautés par Emergence. Montpellier, Université Montpellier 2.

Benson, D. A., I. Karsch-Mizrachi, et al. (2006). "GenBank." Nucleic Acids Research **34**(Database issue): 5.

Bertin, J. (1967). Sémiologie graphique : Les diagrammes, les réseaux, les cartes. Paris, La Haye, Mouton, Gauthier-Villars.

Bertin, J. (1977). La graphique et le traitement graphique de l'information. Paris, Flammarion.

Béthery, A. (2005). Guide de la classification décimale de Dewey : Tables abrégées de la XXIIe édition intégrale en langue anglaise, Cercle De La Librairie.

Bhouwmick, S. S., P. Cruz, et al. (2002). Warehousing and Querying Biological Data using gRNA. International Workshop on Bioinformatics (in conjunction with ISMIS) Lyon (France).

Bianchini, M., M. Gori, et al. (2005). "Inside PageRank." ACM Transactions on Internet Technology (TOIT) **5**(1): 92-128.

Biebow, B. and S. Szulman (1999). Terminae: A Linguistics-Based Tool for the Building of a Domain Ontology. 11th European Workshop on Knowledge Acquisition, Modeling and Management (EKAW'99), Springer.

Bier, E. A., M. C. Stone, et al. (1993). Toolglass and Magic Lenses: The See-through Interface. Computer Graphics Annual Conference Series (ACM SIGGRAPH '93), Anaheim (California, USA), ACM Press.

Birkland, A. and G. Yona (2006). "BIOZON: a system for unification, management and analysis of heterogeneous biological data." BMC Bioinformatics **7**(70).

Bodenreider, O. (2004). "The Unified Medical Language System (UMLS): integrating biomedical terminology." Nucleic Acids Research **32**(Database issue): 267-270.

Bodson, C. (2004). Termes et relations sémantiques en corpus spécialisés : rapport entre patrons de relations sémantiques (PRS) et types sémantiques (TS). . Linguistique et traduction. Montréal (Canada), Faculté des arts et des sciences**:** 298.

Bökman, F. (2001). PAQS - Data Sources, Exchange Formats and Database Schemas for a Proteo-Chemimetric Analysis and Query System. Computer Science. Uppsala (Sweden), Uppsala University. **M.Sc.**

Booch, G., I. Jacobson, et al. (2004). UML 2 : Guide de référence, CampusPress.

Borodin, A., G. O. Roberts, et al. (2005). "Link analysis ranking: algorithms, theory, and experiments." ACM Transactions. On Internet Technology (TOIT) **5**(1): 231-297.

Bourigault, D. (2002). Upery : un outil d'analyse distributionnelle étendue pour la construction d'ontologies à partir de corpus. Traitement Automatique du Langage Naturel (TALN'02).

Bourigault, D., N. Aussenac-Gilles, et al. (2003). "Construction de ressources terminologiques ou ontologiques à partir de textes : un cadre unificateur pour trois études de cas." RIA - Revue d'Intelligence Artificielle: 24.

Bourigault, D. and C. Fabre (2000). Approche linguistique pour l'analyse syntaxique de corpus. Cahiers de Grammaires, Université Toulouse - Le Mirail.

Bozdech, Z., M. Llinás, et al. (2003). "The Transcriptome of the Intraerythrocytic Developmental Cycle of Plasmodium Falciparum." PLoS Biology **1**(1).

Budura, A., P. Cudré-Mauroux, et al. (2007). "From bioinformatic web portals to semantically integrated Data Grid network." Future Generation Computer Systems **23**(3): 485-496.

Bukhman, Y. V. and J. Skolnick (2001). "BioMolQuest: integrated database-based retrieval of protein structural and functional information." Bioinformatics, Oxford University Press **17**(5): 468-478.

Buneman, P., S. Khanna, et al. (2001). Why and Where: A Characterization of Data Provenance. 8th International Conference on Database Theory (ICDT), London (UK), Springer, Lecture Notes in Computer Science.

Burger, E., J. Link, et al. (1997). A Multi-Agent Architecture for the Integration of Genomic Information. 1st Int. Workshop on Intelligent Information Integration (in conjunction with KI'97), Freiburg (Germany).

Buttenfield, B. P. and C. R. Weber (1994). "Proactive graphics for exploratory visualization of biogeographical data." Cartographic Perspectives **19**(3): 8-18.

Calvanese, D., G. De Giacomo, et al. (2001). A Framework for Ontology Integration. The first Semantic Web Working Symposium (SWWS'01), The Emerging Semantic Web, Stanford University (California, USA), IOS Press.

Camon, E., D. Barrell, et al. (2003). "The Gene Ontology Annotation (GOA) Database - An integrated resource of GO annotations to the UniProt Knowledgebase." In Silico Biology **4**.

Camon, E., M. Magrane, et al. (2004). "The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology." Nucleic Acids Research **32**(Database-Issue): 262-266.

Card, S. K., J. Mackinlay, et al. (1999). Readings in Information Visualization: Using Vision to Think, Morgan Kaufmann.

Carpendale, M. S. T. (1999). A Framework for Elastic Presentation Space, Simon Fraser University. **Ph. D**.

Cecchet, E. (2004). "C-JDBC: a Middleware Framework for Database Clustering." IEEE Data Engineering Bulletin **27**(2): 19-26.

Chaffin, R. and D. J. Herrmann (1988). The Nature of Semantic Relations : a Comparison of two Approaches. Relational Models of the Lexicon. Representing Knowledge in Semantic Networks. M. W. Evens. New York (USA), Cambridge University Press**:** 288-334.

Chalmers, M. (1995). Design perspectives in visualizing complex information. IFIP Third Visual Databases Conference. Lausanne (Suisse), Chapman & Hall, Ltd**:** 103-111.

Chamberlin, D. D. and R. F. Boyce (1974). "SEQUEL: A structured English query language." International Conference on Management of Data, Proceedings of the 1974 ACM SIGFIDET (now SIGMOD) workshop on Data description, access and control: 249-264.

Chauché, J. (1990). "Détermination sémantique en analyse structurelle : une expérience basée sur une définition de distance." TAL Information **31**(1): 17-24.

Chawathe, S., H. Garcia-Molina, et al. (1994). The TSIMMIS Project: Integration of Heterogeneous Information Sources. 16th Meeting of the Information Processing Society of Japan, Tokyo (Japan).

Chen, J., P. Zhao, et al. (2004). "The PEPR GeneChip data warehouse, and implementation of a dynamic time series query tool (SGQT) with graphical interface." Nucleic Acids Research **32**(Database-Issue): 578-581.

Chen, P. P.-S. (1976). "The Entity-Relationship Model - Toward a Unified View of Data." ACM Transactions on Database Systems **1**(1): 9-36.

Chen, S., J. Chen, et al. (2004). Detection and correction of conflicting source updates for view maintenance. Proceedings of the 20th International Conference on Data Engineering (ICDE'04), Boston (Massachusetts, USA), IEEE Computer Society.

Chi, E. H.-h. and J. T. Riedl (1998). An Operator Interaction Framework for Visualization Systems. IEEE Symposium on Information Visualization (Infovis'98), IEEE Computer Society.

Chi, E. H. (2002). Expressiveness of the Data Flow and Data State Models in Visualization Systems. Advanced Visual Interfaces., Trento (Italy).

Chiu, S. L. (1997). Extracting fuzzy rules from data for function approximation and pattern classification. H. P. R. R. Y. D. Dubois. New York, Wiley, Fuzzy Information Engineering: a Guide Tour of Applications**:** 149-162.

Church, K. W. and P. Hanks (1991). "Word Association Norms, Mutual Information and Lexicography." Computational Linguistics **16**(1): 22-29.

Codasyl (1971). Codasyl Database Task Group (DBTG) ; April 71 Report. IB-D75224. New York (USA).

Codd, E. F. (1969). "Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks." IBM Research Report **RJ599**.

Codd, E. F. (1970). "A Relational Model of Data for Large Shared Data Banks." CACM **13**(6).

Codd, E. F. (1985). "Does Your DBMS Run By the Rules?" ComputerWorld **21**.

Codd, E. F. (1985). "Is Your DBMS Really Relational?" ComputerWorld **14**.

Cohen-Boulakia, S. (2005). Intégration de données biologiques : sélection de sources centrée sur l'utilisateur. Informatique. Orsay, Université Paris-Sud XI. **Ph.D:** 221.

Cohen-Boulakia, S., O. Biton, et al. (2007). "BioGuideSRS: querying multiple sources with a user-centric perspective." Bioinformatics **23**(10): 1301-1303.

Cohen-Boulakia, S., S. B. Davidson, et al. (2006). "Path-based systems to guide scientists in the maze of biological data sources." Journal of Bioinformatics and Computational Biology **4**(5): 1069-1095.

Collins, A. and R. Quillian (1969). "Retrivial time from semantic memory." Verbal learning and verbal behaviour: 240-247.

Collins, A. and R. Quillian (1970). "Does category size affect categorization time ?" Verbal learning and verbal behaviour: 432-438.

Consortium, G. O. (2001). "Creating the gene ontology resource: design and implementation." Genome Research **11**(8): 1425-33.

Consortium, G. O. (2006). "The Gene Ontology (GO) project in 2006." Nucleic Acids Research **34**(Database Issue): 5.

Consortium, T. U. (2007). "The Universal Protein Resource (UniProt)." Nucleic Acids Research **Vol. 35**(Database issue): 93-197.

Corby, O., R. Dieng-Kuntz, et al. (2006). "Searching the Semantic Web: Approximate Query Processing based on Ontologies." IEEE Intelligent Systems & their Applications **21**(1): 20-27.

Corcho, O., M. Fernández-López, et al. (2003). "Methodologies, tools and languages for building ontologies: where is their meeting point?" Data Knowl. Eng. **46**(1): 41-64.

Cornell, M., N. W. Paton, et al. (2001). GIMS - A Data Warehouse for Storage and Analysis of Genome Sequence and Functional Data. IEEE International Symposium on Bio-Informatics and Biomedical Engineering (BIBE).

Couclelis, H. (1998). "Worlds of information: The geographic metaphor in visualization of complex informatio." Cartography and Geographic Information Systems **25**(4): 209-20.

Coutaz, J. (1990). Interface Homme-Ordinateur : Conception et Réalisation, Dunod.

Coutaz, J. and L. Nigay (2001). Architecture logicielle conceptuelle des systèmes interactifs. Analyse et conception de l'IHM, Interaction pour les systèmes d'information. C. Kolski. Paris, Hermes. **1:** 38.

Crampes, M., S. Ranwez, et al. (2006). An Integrated Visual Approach for Music Indexing and Dynamic Playlist Composition. MultiMedia Computing and Networking Conference (MMCN'06), San José (California, USA), sponsored by ACM Multimedia.

Crampes, M., J. Villerd, et al. (2006). Automatic Playlist Composition in a Dynamic Music Landscape, LGI2P - Ecole des mines d'Alès**:** 6.

Crampes, M., J. Villerd, et al. (2006). Cartes conceptuelles pour l'ingénierie de cartes de connaissances adaptatives. Ingénierie des Connaissances (IC'06), Nantes.

Crochemore, M., C. Hancart, et al. (2001). Algorithmique du texte, Vuibert.

Cunningham, W. and B. Leuf (2001). The Wiki Way. Quick Collaboration on the Web, Addison-Wesley.

Dameron, O. (2003). Modélisation, représentation et partage de connaissances anatomiques sur le cortex cérébral. . Génie biologique et médical. Rennes, Université de Rennes 1. **Ph.D:** 295.

Danchin, A. (1998). La Barque de Delphes. Ce que révèle le texte des génomes, Odile Jacob.

Davidson, S., G. C. Overton, et al. (1995). "Challenges in Integrating Biological Data Sources." Journal of Computational Biology **2**(4): 557-572.

Davidson, S. B., J. Crabtree, et al. (2001). "K2/Kleisli and GUS: Experiments in Integrated Access to Genomic Data Sources." IBM Systems Journal **40**(2): 512-531.

Delobel, C., C. Reynaud, et al. (2003). "Semantic integration in Xyleme: a uniform tree-based approach." Data and Knowledge Engineering **44**(3): 267-298.

DeRisi, J. L., L. R. Vishwanath, et al. (1997). "Exploring the Metabolic and Genetic Control of Gene Expression on a Genomic Scale." Science **279**: 680-686.

Di Battista, G., P. Eades, et al. (1999). Graph Drawing - Algorithms for the Visualization of Graphs, Prentice-Hall.

Ding, Z. and Y. Peng (2004). A Probabilistic Extension to Ontology Language OWL. 37th Hawaii International Conference on System Sciences.

Dodge, M. and R. Kitchin (2001). Atlas of Cyberspace, Addison-Wesley.

Domingue, J. (1998). Tadzebao and webonto: Discussing, browsing, and editing ontologies on the web. 11th Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'98).

Donelson, L., P. Tarczy-Hornoch, et al. (2003). "The BioMediator System as a Data Integration Tool to Answer Diverse Biologic Queries." Medinfo: 768-772.

Durand, P., L. Labarre, et al. (2006). "GenoLink: a graph-based querying and browsing system for investigating the function of genes and proteins." BMC Bioinformatics **7**(21).

Duret, L., E. Gasteiger, et al. (1996). "LALNVIEW: a graphical viewer for pairwise sequence alignments." Comput Applied Bioscience **12**(6): 507-510.

Eades, P. (1984). "A Heuristic for Graph Drawing." Congressus Numerantium **42**: 149-160.

Eckman, B. A., A. S. Kosky, et al. (2001). "Extending traditional query-based integration approaches for functional characterization of post-genomic data." Bioinformatics **17**(7): 587-601.

Eisen, M. B., P. T. Spellman, et al. (1998). "Cluster Analysis and Display of Genome-Wide Expression Patterns." National Academy of Sciences of the United States of America, PNAS **95**(25): 14863-14868.

Embury, S. M. (1994). Constraint-Based Updates in a Functional Data Model Database. Computer Science. Aberdeen, King's College, University of Aberdeen (UK).

Engström, H. and K. Asthorsson (2003). A Data Warehouse Approach to Maintenance of Integrated Biological Data. Workshop on Bioinformatics held in conjunction with IEEE 19th International Conference on Data Engineering (ICDE'03), Bangalore (India).

Escofier, B. and J. Pagès (1988). Analyses factorielles simples et multiples, Dunod.

Etzioni, O. and D. Weld (1994). "A Softbot-based Interface to the Internet." Communications of the ACM **37**(7): 72-76.

Etzold, T., A. Ulyanov, et al. (1996). "SRS: Information Retrieval System for Molecular Biology Data Banks." Methods in Enzymology **266**: 114-128.

Eyre, T. A., F. Ducluzeau, et al. (2006). "The HUGO Gene Nomenclature Database, 2006 updates." Nucleic Acids Research **34**(Database Issue): 319-321.

Fabrikant, S. I. and B. P. Buttenfield (2001). Formalizing Semantic Spaces for Information Access.

Fahl, G., T. Risch, et al. (1993). AMOS - An Architecture for Active Mediators. The International Workshop on Next Generation Information Technologies and Systems (NGITS'93), Technion, Haifa (Israel).

Fall, C. P., E. S. Marland, et al. (2002). Computational Cell Biology - An Introduction to Computer Modeling in Molecular Cell Biology, Springer-Verlag.

Farquhar, A., A. Dappert, et al. (1995). Integrating Information Sources Using Context Logic. AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments, Stanford University (California, USA).

Farquhar, A., R. Fikes, et al. (1995). Collaborative Ontology Construction for Information Integration.

Feng, Z., L. Chen, et al. (2004). "Ligand Depot: a data warehouse for ligands bound to macromolecules." Bioinformatics **20**(13): 2153-2155.

Feuerstein, S. and B. Pribyl (2005). Oracle PL/SQL Programming, O'Reilly.

Fischer, M., Q. K. Thai, et al. (2006). "DWARF - a data warehouse system for analyzing protein families." BMC Bioinformatics **7**(495).

Flanagan, D. (1999). Java™ Foundation Classes in a Nutshell, O'Reilly.

Flower, J. and J. Howse (2002). Generating Euler Diagrams. Diagrams'2002, Springer Verlag.

Flower, J., P. Rodgers, et al. (2003). Layout Metrics for Euler Diagrams. IEEE Information Visualization (IV03).

Freier, A., R. Hofestädt, et al. (2002). "BioDataServer: A SQL-based service for the online integration of life science data." In Silico Biology **2**: 37-57.

Freksa, C. (1999). Spatial aspects of task-specific wayfinding maps. A representation-theoretic perspective. Visual and spatial reasoning in design, Sydney: Key Centre of Design Computing and Cognition.

Friedman, M., A. Y. Levy, et al. (1999). Navigational Plans for Data Integration. 16th National Conference on Artificial Intelligence (AAAI-99),11th Conference on Innovative Applications of Artificial Intelligence (IAAI'99), Orlando (Florida, USA), AAAI/MIT Press.

Friedman, M. and D. S. Weld (1997). "Efficiently Executing Information-Gathering Plans." 15th International Joint Conference on Artificial Intelligence (IJCAI97-1): 785-791.

Fujibuchi, W., S. Goto, et al. (1998). DBGET/LinkDB: an Integrated Database Retrieval System. 3rd Pacific Symposium on Biocomputing.

Furnas, G. W. (1981). The FISHEYE View: A New Look at Structured Files. Murray Hill (New Jersey, USA), Bell Laboratories.

Gallaire, H. and J. Minker (1978). Logic and Databases. New York, Plenum Press.

Galperin, M. Y. (2007). "The Molecular Biology Database Collection: 2007 update." Nucleic Acids Research **35**(Database issue).

Garcia-Molina, H., J. D. Ullman, et al. (2002). Database Systems: The Complete Book, Prentice Hall.

Gardarin, G. (1999). Bases de données, Eyrolles.

Gasch, A., P. and M. Eisen, B. (2002). "Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering." Genome Biology **3**(11): 1-22.

Gasteiger, E., A. Gattiker, et al. (2003). "ExPASy: the proteomics server for in-depth protein knowledge and analysis." Nucleic Acids Research, Oxford University Press **31**(13): 3784-3788.

Gaume, B. (2004). "Balades aléatoires dans les Petits Mondes Lexicaux." I3 - Information - Interaction - Intelligence **4**(2): 59.

Genesereth, M. R. and F. R. E. (1992). Knowledge Interchange Format Version 3.0 Reference Manual, Computer Science Department, Stanford University.

Genesereth, M. R., A. M. Keller, et al. (1997). Infomaster: An Information Integration System. Proceedings of the ACM SIGMOD International Conference on Management of Data (ACM Special Interest Group on Management of Data), Tuscon (Arizona, USA), ACM Press (New York).

Genest, D. and É. Salvat (1998). A Platform Allowing Typed Nested Graphs : How CoGITo Became CoGITaNT. 6th International Conference on Conceptual Structures (ICCS'98), Springer.

Gennari, J. H., M. A. Musen, et al. (2003). "The Evolution of Protégé: An Environment for Knowledge-Based Systems Development." International Journal of Human-Computer Studies, (), pp. , 2003. **58**(1): 89-123.

Gentleman, R., W. Huber, et al. (2005). Bioinformatics And Computational Biology Solutions Using R And Bioconductor, Springer.

Gibson, J. J. (1979). The ecological approach to visual perception. Boston (Massashusetts, USA), Houghton Mifflin.

Gibson, J. J. I.-.-.-. (1977). The Theory of Affordances. In Perceiving, Acting, and Knowing.

Golledge, R. G. and R. J. Stimson (1987). Analytical behavioral geography, Routledge Kegan & Paul.

Goscinny, R., A. Uderzo, et al. (1961). Astérix le gaulois, Hachette.

Gouy, M., F. Milleret, et al. (1984). "ACNUC: a nucleic acid sequence data base and analysis system." Nucleic Acids Research **12**(1): 121-127.

Gray, J. (1981). The transaction concept: Virtues and limitations. 7th International Conference on Very Large Data Bases. Cannes (France), Tandem Computers Inc.**:** 144-154.

Gribble, S. D., A. Y. Halevy, et al. (2001). What Can Database Do for Peer-to-Peer? International Workshop on the Web and Databases (WebDB'2001), Santa Barbara (California, USA).

Gruber, T. R. (1993). "A translation approach to portable ontology specifications." Knowledge Acquisition **5**(2): 199-220.

Guarino, N. (1994). The Ontological Level. Philosophy and the Cognitive Sciences. O. Casati, B. Smith and G. White. Vienne (Autriche), Hölder-Pichler-Tempsky.

Guérin, E., G. Marquet, et al. (2005). Integrating and Warehousing Liver Gene Expression Data and Related Biomedical Resources in GEDAW. Second International Workshop on Data Integration in the Life Sciences, San Diego (CA, USA), Springer.

Guindon, S. (2003). Méthodes et algorithmes pour l'approche statistique en phylogénie. Sciences Chimiques et Biologiques pour la Santé. Montpellier, Université Montpellier II. **Ph.D.:** 155.

Gupta, A., B. Ludascher, et al. (2003). BIRN-M: A Semantic Mediator for Solving Real-World Neuroscience Problems. ACM SIGMOD international conference on Management of data, Demonstration Session: Potpourri, San Diego (California, USA), ACM Press.

Gupta, A., B. Ludäscher, et al. (2002). Registering Scientific Information Sources for Semantic Mediation. 21st International Conference on Conceptual Modeling, Tampere (Finland), Springer.

Gupta, A. and I. S. Mumick (1995). "Maintenance of Materialized Views: Problems, Techniques and Applications." IEEE Quarterly Bulletin on Data Engineering; Special Issue on Materialized Views and Data Warehousing **18**(2): 3-18.

Gómez, A., A. Moreno, et al. (2000). "Knowledge Maps: An essential technique for conceptualisation." Data & Knowledge Engineering **33**(2): 169-190.

Haarslev, V. and R. Möller (2001). RACER System Description. Automated Reasoning : First International Joint Conference, IJCAR 2001, Lecture Notes in Computer Science, Springer.

Haas, L. M., P. M. Schwarz, et al. (2001). "DiscoveryLink: A system for integrated access to life sciences data sources " IBM Systems Journal, Deep computing for the life sciences **40**(2): 489-511.

Halpin, T. (2001). Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design, Morgan Kaufmann.

Harris, Z. S., M. Gottfried, et al. (1989). "The form of Information in Science, Analysis of Immunology Sublanguage." Boston Studies in the Philosophy of Science, Kluwer Academic Publisher, Dordrecht **104**: 590.

Hauser, H., F. Ledermann, et al. (2002). Angular Brushing for Extended Parallel Coordinates. IEEE Symposium on Information Visualization 2002 (InfoVis 2002), Boston (Massachusetts, USA).

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. Computational Linguistics (Coling'92), Nantes (France).

Heer, J. and D. Boyd (2005). Vizster: Visualizing Online Social Networks. IEEE Symposium on Information Visualization (2005).

Heer, J., S. K. Card, et al. (2005). prefuse: a toolkit for interactive information visualization. Human Factors in Computing Systems (CHI 2005).

Hernandez, T. and S. Kambhampati (2004). "Integration of Biological Sources: Current Systems and Challenges Ahead." SIGMOD Record (ACM Special Interest Group on Management of Data) **33**(3): 51-60.

Hertz-Fowler, C., C. S. Peacock, et al. (2004). "GeneDB: a resource for prokaryotic and eukaryotic organisms." Nucleic Acids Research **32**(Database Issue): 339-343.

Horrocks, I. and U. Sattler (2001). Ontology Reasoning in the SHOQ(D) Description Logic. IJCAI.

Hull, D., K. Wolstencroft, et al. (2006). "Taverna: a tool for building and running workflows of services." Nucleic Acids Research, Oxford University Press **34**(Web Server issue): W729-W732.

Iazzetti, G., G. Santini, et al. (1998). "VIRTLAB: a virtual molecular biology laboratory." Bioinformatics **14**(9): 815-816.

Ihaka, R. and R. Gentleman (1996). "R: A language for data analysis and graphics." Computational and Graphical Statistics **5**(3): 299-314.

Inmon, W. H. (1992). Building the Data Warehouse. New York, Wiley Computer Pub.

Inselberg, A. (1985). "The plane with parallel coordinates." The Visual Computer **1**(2): 69-91.

ISO ISO/IEC 13211: Information technology - Programming languages - Prolog. Geneva, International Organization for Standardization.

Ives, Z. G., A. Y. Halevy, et al. (2004). "Piazza: mediation and integration infrastructure for Semantic Web data." Journal of Web Semantics **2**(1): 155-175.

Jalabert, F. (2003). Catégorisation de définitions et nommage de sens, Université Montpellier 2. **Master**.

Jalabert, F., B. Munier, et al. (2002). Correcteur orthographique. T. Maîtrise, Université Montpellier 2.

Jalabert, F., S. Ranwez, et al. (2005). Médiation et environnement intégré d'ingénierie ontologique. Ingénierie des connaissances (IC'05) - Session poster.

Jalabert, F., S. Ranwez, et al. (2006). I²DEE : un environnement intégré pour l'exploration de données biologiques**:** 12.

Jalabert, F., S. Ranwez, et al. (2006). I2DEE: an Integrated and Interactive Data Exploration Environment used for Ontology Design. 15th International Conference on Knowledge Engineering and Knowledge Management Managing Knowledge in a World of Networks (EKAW 2006), Podebrady (Czech Republic), Springer Verlag.

Jenssen, T.-K., A. Laereid, et al. (2001). "A Literature network of human genes for high-throughput analysis of gene expression." Nature Genetics **28**(1): 21-28.

Ji, H. and S. Ploux (2003). Automatic contexonym organizing model. 25th annual meeting of the Cognitive Science Society.

Johnson, M. (1987). The body in the mind: Bodily basis of meaning, imagination, and reason. Chicago, University of Chicago Press.

Juola, J. F. and R. C. Atkinson (1971). "Memory scanning for word versus categories." Journal of verbal learning and verbal behaviour: 449-452.

Karolchik, D., R. Baertsch, et al. (2003). "The UCSC Genome Browser Database." Nucleic Acids Research **31**(1): 51-54.

Kasprzyk, A., D. Keefe, et al. (2004). "EnsMart: A Generic System for Fast and Flexible Access to Biological Data." Genome Research **14**(1): 160-169.

Kato, K., R. Yamashita, et al. (2005). "Cancer gene expression database (CGED): a database for gene expression profiling with accompanying clinical information of human cancer tissues." Nucleic Acids Research **33**(Database Issue): 533-536.

Kawas, E., M. Senger, et al. (2006). "BioMoby extensions to the Taverna workflow management and enactment software." BMC Bioinformatics **7**(523).

Kifer, M., G. Lausen, et al. (1995). "Logical Foundations of Object-Oriented and Frame-Based Languages." Journal of the Association for Computing Machinery **42**(4): 741-843.

Kimball, R. and M. Ross (1996). The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, John Wiley & Sons.

Kirk, T., A. Y. Levy, et al. (1995). The Information Manifold. Information Gathering from Heterogeneous, Distributed Environments - AAAI Spring Symposium on Information Gathering, Stanford University (California, USA).

Kleinberg, J. M. (1998). Authoritative sources in a hyperlinked environment. 9th annual ACM-SIAM symposium on discrete algorithms (SODA'98), San Francisco (California, USA), Society for Industrial and Applied Mathematics.

Klippel, A., P. U. Lee, et al. (2005). The Cognitive Conceptual Approach as a Leitmotif for Map Design. AAAI Spring Symposium in Reasoning with Mental and External Diagrams: Computational Modeling and Spatial Assistance. Stanford (California, USA), AAAI Press.

Klippel, A. and H. Tappe (2001). Conceptual spacial representations in language an graphics. Cognitive Systems & Mechanisms (KogWis 2001), 5th meeting of German Cognitive Science Society.

Köhler, J., S. Philippi, et al. (2003). "SEMEDA: ontology based semantic integration of biological databases." Bioinformatics **19**(18): 2420-2427.

Kohler, J. and S. Schulze-Kremer (2002). "The semantic metadatabase (SEMEDA): ontology based integration of federated molecular biological data sources." In Silico Biology **2**(3): 219-231.

Kohonen, T. (1990). "The self-organizing map." IEEE **78**(9): 1464-1480.

Koss, M. (2002). Apollo - The user guide 1.0, Knowledge Media Institute, The Open University.

Kostoff, R. N., J. A. Block, et al. (2004). "Information content in Medline record fields." International Journal of Medical Informatics **73**(6): 515-527.

Kouranov, A., L. Xie, et al. (2006). "The RCSB PDB information portal for structural genomics." Nucleic Acids Research **34**(Database Issue): 4.

Kozaki, K., Y. Kitamura, et al. (2002). "Hozo: An Environment for Building/Using Ontologies Based on a Fundamental Consideration of "Role" and "Relationship"." 213-218.

Krasner, G. and S. Pope (1988). "A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80." ournal of Object Oriented Programming **26-49**.

Lacroix, Z., K. Parekh, et al. (2005). BioNavigation: Using Ontologies to Express Meaningful Navigational Queries Over Biological Resources. Fourth International IEEE Computer Society Computational Systems Bioinformatics Conference Workshops (CSB 2005 Workshops), Stanford (California, USA), IEEE Computer Society.

Lafourcade, M., V. Prince, et al. (2002). "Vecteurs conceptuels et structuration émergente de terminologies " Revue TAL **43**(1): 43-72.

Landauer, T. and J. Freedman (1968). "Information Retrivial from long term memory : category size and recognition." Journal of verbal learning and verbal behaviour: 291-331.

Le Ber, F. and A. Napoli (2005). "Relations, structures et objets: quelques variations." RSTI L'objet, Hermès (Paris) **11**(1-2): 17-190.

Le Bihan, F., J.-L. Deladrière, et al. (2004). Organisez vos idées avec le Mind Mapping, Dunod.

Le Grand, B. (2001). Extraction d'information et visualisation de systèmes complexes sémantiquement structurés. Systèmes informatiques. Paris, l'Université Pierre et Marie Curie, Paris VI. **Ph.D:** 172.

Lee, T. J., Y. Pouliot, et al. (2006). "BioWarehouse: a bioinformatics database warehouse toolkit." BMC Bioinformatics **7**(170).

Leonard, H. S. and N. Goodman (1940). "The Calculus of Individuals and Its Uses." Journal of Symbolic Logic **5**: 45-55.

Leśniewski, S. (1927). Sur les fondements de la mathématique. Paris, Hermès.

Letondal, C. (2001). Interaction et Programmation. Computer Science. Orsay, Université de Paris XI. **Ph. D**.

Levy, A. Y. (1999). Combining Artificial Intelligence and Databases for Data Integration. Artificial Intelligence Today: Recent Trends and Developments. Berlin / Heidelberg, Springer **1600**.

Li, L., J. Crabtree, et al. (2004). "ApiEST-DB: analyzing clustered EST data of the apicomplexan parasites." Nucleic Acids Research **32**(Database issue): 326-328.

Lin, W. (2004). Applications de la technologie des Puces à ADN à l'étude de la différenciation méiotique et des mécanismes de recombinaison chez la levure Saccharomyces cerevisiae. UMR144 CNRS - Institut Curie. Paris, Université Paris 6. **Ph.D**.

Lister, R., M. W. Murcha, et al. (2003). "The Mitochondrial Protein Import Machinery of Plants (MPIMP) database." Nucleic Acids Research **31**(1): 325-327.

Lopez, M. F., A. Gomez-Perez, et al. (1999). "Building a chemical ontology using Methontology and the ontology design environment." Intelligent Systems, IEEE **14**(1): 37-46.

MacEachrean, A. M. (1995). How maps work: Representation, visualization, and design, The Guilford press.

Mackinlay, J. D. (1986). "Automating the Design of Graphical Presentations of Relational Information." ACM Transactions on Graphics **5**(2): 110-141.

Maedche, A. (2002). Ontology Learning for the Semantic Web, Kluwer Academic.

Maglott, D. R., J. Ostell, et al. (2005). "Entrez Gene: gene-centered information at NCBI." Nucleic Acids Research **33**(Database issue): 54-58.

Mangeot-Lerebours, M. (2001). Environnements centralisés et distribués pour lexicographes et lexicologues en contexte multilingue. Informatique. Grenoble (France), Université Joseph Fourier. **Ph. D**.

Maniez, J. (2005). Actualités des langages documentaires ; fondements théoriques de la recherche d'information, ADBS Editions.

Manning, C. D. and H. Schütze (1999). Foundations of Statistical Natural Language Processing. Cambridge, Massachusetts, The MIT Press.

Marchiori, M. (1997). The quest for correct information on the Web: Hyper search engines. 6th International World-Wide Web Conference, Santa Clara (California, USA).

Mark, D. M. and A. U. Frank (1991). Cognitive and linguistic aspects of geographic space. Dortrecht (The Netherlands), Kluwer Academic Publishers.

Marshman, E., T. Morgan, et al. (2002). "French patterns for expressing concept relations." Terminology **8**(1): 1-29.

Martin, S., M. M. Hohman, et al. (2005). "The impact of Life Science Identifier on informatics data." Drug Discovery Today **10**(22): 1566-1572.

McBride, B. (2001). Jena: Implementing the RDF Model and Syntax Specification. The Second International Workshop on the Semantic Web (SemWeb'2001) - World Wide Web Conference 2001(WWW2001), Hong-Kong (China).

McGuinness, D. L. and F. van Harmelen (2004). OWL Web Ontology Language - Overview, W3C Recommendation.

Mcilwaine, I. C. (2000). The Universal Decimal Classification: guide to its use, The Hague : UDC Consortium.

McKusick, V. A. (1998). Mendelian Inheritance in Man ; A Catalog of Human Genes and Genetic Disorders. 12th Edition. Baltimore, The Johns Hopkins University Press.

McMaster, R. and E. L. Usery (2004). A Research Agenda for Geographic Information Science, CRC Press.

Médigue, C., F. Rechenmann, et al. (1999). "Imagene: an integrated computer environment for sequence annotation and analysis." Bioinformatics **15**(1): 2-15.

Mel'čuk, I. (1988). Dictionnaire explicatif et combinatoire du français contemporain, volume 2. . Montréal, Les presses de L'université de Montréal.

Mena, E., V. Kashyap, et al. (1996). OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies 4th International Conference on Cooperative Information Systems, Bruessels (Belgium).

Michalickova, K., G. D. Bader, et al. (2002). "SeqHound: biological sequence and structure database as a platform for bioinformatics research." BMC Bioinformatics **3**(32).

Michel Biezunski, M. B. and S. R. Newcomb (1999). ISO/IEC 13250 - Topic Maps.

Miller, G. A. and W. G. Charles (1991). "Contextual Correlates of Semantic Similarity." Language and Cognitive Processes **6**(1): 1-28.

Miranda, S. (2002). Bases de données - Architectures, modèles relationnels et objets, SQL 3, Dunod.

Mizoguchi, R. (2004). Le rôle de l'ingénierie ontologique dans le domaine des EIAH, entretien réalisé par Jacqueline Bourdeau.

Mizoguchi, R. (2004). Ontology engineering environments. Handbook on Ontologies. S. Staab and R. Studer, Springer-Verlag**:** 275-298.

Mokrane, A. (2006). Représentation de collections de documents textuels : Application à la caractérisation thématique. Informatique. Nimes, Université Montpellier II. **Ph.D:** 115.

Muehrcke, P. C., J. O. Muehrcke, et al. (2001). Map Use: Reading, Analysis, Interpretation. , Madison: J.P. Publications.

Munzner, T. (2000). Interactive visualization of large graphs and networks, Stanford University.

Neyfakh, A., A., N. Baranova, N., et al. (2006). "A system for studying evolution of life-like virtual organisms." Biology Direct **1**(23): 21.

Nigay, L. (1994). Conception et modélisation logicielles des systèmes interactifs : application aux interfaces multimodales. Grenoble, Université Joseph Fourier.

Nijssen, G. M. and T. A. Halpin (1989). Conceptual Schema and Relational Database Design. Sydney, Prentice Hall.

Nikitin, A., S. Egorov, et al. (2003). "Pathway studio - the analysis and navigation of molecular networks." Bioinformatics **19**(16): 2155-2157.

Norman, D. A. The Design of Everyday Things.

Noy, N. F. (1997). Knowledge representation for intelligent information retrieval in experimental sciences, Northeastern University.

Nyckees, V. (1998). La s´emantique, Belin.

Oberle, D., R. Volz, et al. (2004). An extensible ontology software environment. Handbook on Ontologies. S. Staab and R. Studer, Springer**:** 311-333.

Page, L., S. Brin, et al. (1998). The PageRank Citation Ranking: Bringing Order to the Web, Stanford Digital Library Technologies Project.

Page, M., J. Gensel, et al. (2000). Représentation de connaissances au moyen de classes et d'associations: le système AROM. Actes des journées Langages et Modèles à Objets (LMO'00), Mont Saint-Hilaire (Québec), Hermès.

Park, J. and S. Hunting (2002). Xml Topic Maps: Creating and Using Topic Maps for the Web, Addison-Wesley Professional

Parkinson, H., M. Kapushesky, et al. (2007). "ArrayExpress - a public database of microarray experiments and gene expression profiles." Nucleic Acids Research **35**(Database issue): 747-750.

Patry, G. (1999). Contribution à la conception du dialogue Homme Machine dans les applications graphiques interactives de conception technique : le système GIPSE, Université de Poitier.

Pechoin, D. (1999). Thesaurus : des idées aux mots, des mots aux idées ... , Larousse.

Perrière, G., C. Combet, et al. (2003). "Integrated databanks access and sequence/structure analysis services at the PBIL." Nucleic Acids Research, Oxford University Press **31**(13): 3393-3399.

Pfaff, G. E. (1985). User Interface Management Systems, Springer-Verlag.

Pierret, J.-D. and E. Boutin (2004). "Découverte de connaissances dans les bases de données bibliographiques - Le travail de Don Swanson : de l'idée au modèle." **12**: 7.

Pook, S. and E. Lecolinet (2002). " Interfaces zoomables et Control menus : Techniques focus+contexte pour la navigation interactive dans les bases de donnees." Revue les Cahiers du numerique **3**: 191-210.

Porter, M. F. (1980). "An algorithm for suffix stripping." Program **14**(3): 8.

Potamias, G. (2006). State of the Art on Systems for Data Analysis, Information Retrieval and Decision Support, the Infobiomed Consortium - Information Society Technologies (IST).

Pottier, B. (1964). "Vers une sémantique moderne." Travaux de sémantique et de littérature: 107-137.

Prince, V. (1991). Notes sur l'évaluation de la réponse dans TEDDI : introduction d'une relation d'équivalence pour la synonymie relative. Notes et Documents LIMSI-CNRS.

Pruitt, K. D. and D. R. Maglott (2001). "RefSeq and LocusLink: NCBI gene-centered resources." Nucleic Acids Research **29**(1): 137-140.

Pruitt, K. D., T. Tatusova, et al. (2005). "NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins." Nucleic Acids Research **33**(Database issue): 501-504.

Rahm, E., T. Kirsten, et al. (2007). "The GeWare data warehouse platform for the analysis of molecular-biological and clinical data." Journal of Integrative Bioinformatics **4**(1).

Ranwez, S., V. Ranwez, et al. (2006). Ontological ISA-Distance Measure for Information Visualisation on Conceptual Maps. On the Move to Meaningful Internet Systems (OTM 2006 Workshops), Springer Berlin / Heidelberg.

Rastier, F. (1995). "Le terme : entre ontologie et linguistique." La banque des mots **7**: 35-65.

Rebhan, M., V. Chalifa-Caspi, et al. (1998). "GeneCards: a novel functional genomics compendium with automated data mining and query reformulation support." Bioinformatics **14**(8): 656-664.

Rhee, S. Y., W. Beavis, et al. (2003). "The Arabidopsis Information Resource (TAIR): a model organism database providing a centralized, curated gateway to Arabidopsis biology, research materials and community." Nucleic Acids Research **31**(1): 224-228.

Röber, N. (2000). Multidimensional analysis and visualization software for dynamic SPECT. Magdeburg, Otto-von-Guericke-Universitat. **Master**.

Robertson, G., M. Czerwinski, et al. (1998). Data Mountain:

Using Spatial Memory for Document Management. 11th annual symposium on User Interface Software and Technology (UIST'98). San Francisco (California, USA), Sponsored by ACM SIGGRAPH and SIGCHI and in cooperation with ACM SIGSOFT. .

Robertson, N., M. Oveisi-Fordorei, et al. (2007). "DiscoverySpace: an interactive data analysis application." Genome Biology **8**(R6).

Rocchio, J. J. (1971). Relevance feedback in information retrieval. The Smart retrieval system - experiments in automatic document processing. G. Salton, Englewood Cliffs, NJ: Prentice-Hall**:** 313-323.

Rochfeld, A. and J. Morejon (1989). La méthode Merise - Tome 3 Gamme opératoire. Paris, Editions d'organisation.

Roget, P. M. (1852). Thesaurus of English Words and Phrases. London, Longman.

Safran, M., I. Solomon, et al. (2002). "GeneCardsTM 2002: towards a complete, object-oriented, human gene compendium." Bioinformatics **18**(11): 1542-1543.

Sager, J. C. (1990). A practical Course in Terminology Processing. Amsterdam/Philadelphia, John Benjamins.

Salton, G. and C. Buckley (1987). Term-weighting approaches in automatic text retrieval, Cornell University.

Saporta, G. (2006). Probabilités, analyses des données et statistiques, Technip.

Sarkar, M. and M. H. Brown (1992). Graphical Fisheye Views of Graphs. Human Factors in Computing Systems (CHI'92) - ACM/SIGCHI, Monterey (California, USA), ACM Press.

Sarry, J.-E., L. Kuhn, et al. (2006). "The early responses of Arabidopsis thaliana cels to cadmium exposure explored by protein and metabolite profiling analyses." Proteomics **6**: 2180-2198.

Schulman, J.-L. (1997). "MeSH on the Web." National Library of Medicine Technical Bulletin.

Schwab, D. (2005). Approche hybride - lexicale et thématique - pour la modélisation, la détection et l'exploitation des fonctions lexicales en vue de l'analyse sémantique de texte. Informatique. Montpellier, Université Montpellier 2. **Ph.D:** 390.

Schwab, D., M. Lafourcade, et al. (2002). Vers l'apprentissage automatique pour et par les vecteurs conceptuels de fonctions lexicales - l'exemple de l'antonymie. Traitement Automatique du Langage Naturel (TALN'2002),, Nancy (France).

Seo, J. and B. Shneiderman (2002). "Interactively Exploring Hierarchical Clustering Results." IEEE Computer **35**(7): 80-86.

Shah, S., Y. Huang, et al. (2005). "Atlas - a data warehouse for integrative bioinformatics." BMC Bioinformatics **6**(34).

Shapiro, A. (2003). "TouchGraph." from http://www.touchgraph.com.

Shetty, R. T. N., P.-M. Riccio, et al. (2006). "Hybrid Model for Knowledge Representation (ICHIT'06) " International Conference on Hybrid Information Technology - Vol1 (ICHIT'06) 355-361.

Shneiderman, B. (1987). Designing the User Interface: Strategies for Effective Human-Computer Interaction, Addison-Wesley.

Shneiderman, B. (1992). Tree Visualization with Tree-maps: A 2-D space-filling approach. ACM Transactions on Graphics.

Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualization. IEEE Workshop on Visual Languages.

Siepel, A., A. Farmer, et al. (2001). "ISYS: a decentralized, component-based approach to the integration of heterogeneous bioinformatics resources." Bioinformatics, Oxford University Press **17**(1): 83-94.

Silverstein, K. A. T., E. Shoop, et al. (2001). "The MetaFam Server: a comprehensive protein family resource." Nucleic Acids Research **29**(1): 49-51.

Sirin, E., B. Parsia, et al. (2006). "Pellet: A practical OWL-DL reasoner." Journal of Web Semantics.

Skupin, A. and B. P. Buttenfield (1997). Spatial Metaphore for Visualizing Information Spaces. AUTO-CARTO 13. Bethesda, ACSM/ASPRS**:** 116-125.

Southern, E. M. (1975). "Detection of specific sequences among DNA fragments separated by gel electrophoresis." Molecular Biology **98**(3): 503-517.

Sowa, J. F. (1984). Conceptual Structures: Information Processing in Minds and Machines, Addison-Wesley.

Sowa, J. F. (1995). "The role of formal ontology in the information technnology." International journal of human-computer studies - Elsevier (Londre, UK) **43**(5/6): 669-685.

Stapley, B. J. and G. Benoit (2000). BioBiblioMetrics: information retrieval and visualization from co-occurrences of gene names in Medline abstracts. Pacific Symposia in Biocomputing.

Stein, L., P. Sternberg, et al. (2001). "WormBase: network access to the genome and biology of Caenorhabditis elegans." Nucleic Acids Research, Oxford University Press **29**(1): 82-86.

Stein, L. D. (2003). " Integrating biological databases." Nature Reviews Genetics **4**: 337-345.

Stevens, R. D., A. J. Robinson, et al. (2003). myGrid: personalised bioinformatics on the information grid. Proceedings of the Eleventh International Conference on Intelligent Systems for Molecular Biology (ISMB'03)2003, Brisbane (Australia).

Stock, O. (1998). Spatial And Temporal Reasoning, Springer.

Strahler, A. N. (1952). "Hypsometric (area-altitude) analysis of erosional topology." Bulletin of the Geological Society of America **63**: 1117–1142.

Subrahmanian, V. S., S. Adali, et al. (1995). HERMES: A heterogeneous reasoning and mediator system, University of Maryland.

Subramanian, A., P. Tamayo, et al. (2005). "Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles." Natl Acad Sci U S A. **102**(43): 15545-15550.

Sujansky, W. (2001). "Heterogeneous Database Integration in Biomedicine." Journal of Biomedical Informatics **34**(4): 285-298.

Sure, Y., M. Erdmann, et al. (2002). OntoEdit: Collaborative Ontology Development for the Semantic Web. International Semantic Web Conference.

Sure, Y., S. Staab, et al. (2004). On-To-Knowledge Methodology (OTKM). Handbook on Ontologies**:** 117-132.

Swanson, D. R. (1986). "Fish oil, Raynaud's syndrome, and undiscovered public knowledge." Perspect Biol Med. **30**(1): 7-18.

Swartout, B., R. Patil, et al. (1996). Toward distributed use of large-scale ontologies. Symposium on Ontological Engineering of AAAI. University of South California / Information Sciences Institute (California, USA).

Tardieu, H., A. Rochfeld, et al. (1983). La méthode Merise - Tome 1 Principes et outils. Paris, Editions d'organisation.

Tardieu, H., A. Rochfeld, et al. (1985). La méthode Merise - Tome 2 Démarches et pratiques. Paris, Editions d'organisation.

Tatarinov, I. and A. Y. Halevy (2004). Efficient Query Reformulation in Peer Data Management Systems. Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data 2004, Paris (France), ACM Press.

Taylor, H. A. and B. Tversky (1996). "Perspective in spatial descriptions." Journal of Memory and Language **35**(3): 371-391.

The Plasmodium Genome Database Collaborative (2001). "PlasmoDB: an integrative database of the Plasmodium falciparum genome. Tools for accessing and analyzing finished and unfinished sequence data." Nucleic Acids Research **29**(1): 66-69.

Tomasic, A., L. Raschid, et al. (1996). Scaling Heterogeneous Databases and the Design of DISCO. Proceedings of the 16th International Conference on Distributed Computing Systems (ICDCS'96), Hong Kong (China), IEEE Computer Society.

Tomita, M., K. Hashimoto, et al. (1999). "E-CELL: Software Environment for Whole Cell Simulation." Bioinformatics **15**(1): 72-84.

Tork Roth, M., M. Arya, et al. (1996). The Garlic project. Proceedings of the 1996 ACM SIGMOD (Special Interest Group on Management of Data) International

Conference on Management of Data, Montreal (Quebec, Canada), ACM Press.

Tricot, C. (2006). Cartographie sémantique - Des connaissances à la carte. Annecy le Vieux, Université de Savoie.

Troncy, R. and A. Isaac (2002). DOE : une mise en oeuvre d'une méthode de structuration différentielle pour les ontologies. 13ièmes Journées Francophones d'Ingénierie des Connaissances, IC'2002. Rouen (France).

Tsarkov, D. and I. Horrocks (2006). FaCT++ Description Logic Reasoner: System Description. International Joint Conference on Automated Reasoning (IJCAR 2006), Lecture Notes in Artificial Intelligence, Springer, . **4130** 292-297.

Tsichritzis, D. and F. H. Lochovsky (1976). "Hierarchical Database Management: A Survey." ACM Computing Surveys **8**(1): 105-124.

Tufte, E. (1983). The Visual Display of Quantative Information, Graphics press.

Tufte, E. (1990). Envisioning Information, Graphics Press.

Tufte, E. (1997). Visual Explanations: Images and Quantities, Evidence and Narrative, Graphics Press.

Tullis, T. S. (1985). Designing a Menu-based Interface to an Operating System. CHI'85 Conference on Human Factors in Computing Systems. San Francisco, Calfornia, USA.

Tversky, B. (1995). Cognitive Origins of Graphic Conventions. Understanding Images. F. T. Marchese. New York, Springer-Verlag**:** 29-53.

Van Campenhoudt, M. (1994). Un apport du monde maritime à la terminologie notionnelle multilingue. Etude du dictionnaire du capitaine Heinrich Paasch « De la quille à la pomme de mât » (1885-1901). Sciences du langage. Paris, Université de Paris XIII. **Ph. D:** 431.

Varzi, A. C. (1996). "Parts, Wholes, and Part-Whole Relations: The Prospects of Mereotopology." Data Knowledge Engineering **20**(3): 259-286.

Varzi, A. C. (1996). "Reasoning About Space: The Hole Story." Logic and Logical Philosophy **4**: 3-39.

Véronis, J. (2004). "Hyperlex : lexical cartography for information retrieval." Computer, Speech and Language **18**(3): 223-252.

Vert, J.-P. (2004). Kernel Methods in Computational Biology. Mathématiques. Paris, University Paris 6.

Wain, H. M., M. J. Lush, et al. (2002). "Genew: the Human Gene Nomenclature Database." Nucleic Acids Research **30**(1): 169-171.

Ware, C. (2000). Information Visualization: Perception for Design, Morgan Kaufmann.

Watanabe, J., H. Wakaguri, et al. (2007). "Comparasite: a database for comparative study of transcriptomes of parasites defined by full-length cDNAs." Nucleic Acids Research **35**(Database Issue): 431-438.

Wheeler, D. L., T. Barrett, et al. (2006). "Database resources of the National Center for Biotechnology Information." Nucleic Acids Research **34**(Database Issue): 8.

Wiederhold, G. (1992). Mediators in the Architecture of Future Information Systems. EEE Computer. M. N. Huhns and M. P. Singh. San Francisco (CA, USA), Morgan Kaufmann. **25:** 38-49.

Wikipedia. "Connexionisme." from http://fr.wikipedia.org/wiki/Connexionisme.

Winston, M. E., R. Chaffin, et al. (1987). "A Taxonomy of Part-Whole Relations." Cognitive Science **11**(4): 417-444.

Wroe, C. J., R. D. Stevens, et al. (2003). A Methodology to Migrate the Gene Ontology to a Description Logic Environment Using DAML+OIL. In 8th Pacific Symposium on biocomputing (PSB).

XTM (2001). XML Topic Maps (XTM) 1.0. S. Pepper and G. Moore, TopicMaps.Org Authoring Group.

Zdobnov, E. M., R. Lopez, et al. (2002). "The EBI SRS server: recent developments." Bioinformatics **18**(2): 368-373.

Zuyderduyn, S. D. and S. J. Jones (2003). "A knowledge discovery object model API for Java." BMC Bioinformatics **4**(51).

Zweigenbaum, P. (1999). "Encoder l'information médicale : des terminologies aux systèmes de représentation des connaissances." Innovation Stratégique en Information de Santé **2**(3): 27-47.

Références en ligne

ABU la bibliothèque universelle – *http://abu.cnam.fr/DICO/*

AceDB *http://www.acedb.org/*

Acnuc *http://pbil.univ-lyon1.fr/databases/acnuc/acnuc.html*

Affymetrix *http://www.affymetrix.com/index.affx*

AFNOR Association Française de NORmalisation – *http://www.afnor.org/portail.asp*

Agilent *http://www.home.agilent.com*

Amadea *http://www.isoft.fr/html/prod_amadea.htm*

ApiDB Apicomplexan Database Resource – *http://apidb.org/apidb/*

ApiDots (anciennement ApiEST) – *http://www.cbil.upenn.edu/apidots/*

Apollo *http://apollo.open.ac.uk/*

Applied BioSystems *http://www.appliedbiosystems.com/*

ArrayExpress *http://www.ebi.ac.uk/arrayexpress*

ASN.1 *http://www.asn1.org/*

Atlas *http://bioinformatics.ubc.ca/atlas/*

Atlas of Cyberspace *http://www.cybergeography.org/atlas/atlas.html*

Baciis Biological and Chemical Information Integration System – *http://baciis.engr.iupui.edu/*

Berkeley DB (Oracle) *http://www.oracle.com/technology/products/berkeley-db/index.html*

BioBroker *http://uranos.khaos.uma.es/mediator/*

BioConductor *http://www.bioconductor.org/*

BiodataServer *http://integration.genophen.de/*

Bioguide *http://bioguide-project.net/*

Bioinformatics *http://bioinformatics.oxfordjournals.org/*

Biomediator *http://www.biomediator.org/*

BioMoby *http://www.biomoby.org/*

BioNavigation *http://bioinformatics.eas.asu.edu/BioNavigation.html*

BioWarehouse *http://biowarehouse.ai.sri.com/*

Biozon *http://biozon.org*

BLIMP Biomedical LIterature (and text) Mining Publications – *http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470094397,descCd-tableOfContents.html*

BMC Biomed Central – *http://www.biomedcentral.com/*

Cartactive GDR– *http://cassini.univ-lr.fr/versionFrancaise/GDR.html*

CEA Commissaria à l'énergie atomique – *http://www.cea.fr/*

CGED Cancer Gene Expression Database – *http://cged.genes.nig.ac.jp/*

CGI Common Gateway Interface – *http://hoohoo.ncsa.uiuc.edu/cgi/*

CharGer *http://www.cs.uah.edu/~delugach/CharGer/body.html*

C-JDBC Clustered JDBC – *http://c-jdbc.objectweb.org/*

CNRS Centre national de la Recherche Scientifique – *http://www.cnrs.fr/*

CoGITaNT *http://cogitant.sourceforge.net/*

Comparasite Full-length cDNA Database – *http://fullmal.hgc.jp/comp_index.html*

CORBA Common Object Request Broker Architecture – *http://www.omg.org/*

Cordial (Synapse) http://www.synapse-fr.com/Cordial_Analyseur/Presention_Cordial_Analyseur.htm

CoryneRegNet https://www.cebitec.uni-bielefeld.de/groups/gi/software/coryneregnet/v3/

CSISS Center for Spatially Integrated Social Science – http://www.csiss.org/

CSS Cascading Style Sheets – http://www.w3.org/TR/REC-CSS2/

DAGEdit http://www.godatabase.org/dev/java/dagedit/docs/index.html

DataLog http://en.wikipedia.org/wiki/Datalog

DAVID Database for Annotation, Visualization and Integrated Discovery – http://david.niaid.nih.gov/david/version2/index.htm

DDBJ DNA Data Bank of Japan – http://www.ddbj.nig.ac.jp/

Derby (Apache) http://db.apache.org/derby/

DFDL http://www.gnu.org/copyleft/fdl.html

DiscoverySpace http://www.bcgsc.ca/discoveryspace/

DSSSL Document Style Semantics and Specification Language – http://dsssl.netfolder.com/

DUET http://codip.grci.com/codipsite/codipsite/index.html

DUET http://codip.grci.com/wwwlibrary/DUET_Docs/index.html

Dwarf http://www.led.uni-stuttgart.de/

EBI European Bioinformatics Institute – http://www.ebi.ac.uk/

EC Enzyme Nomenclature – http://www.chem.qmul.ac.uk/iubmb/enzyme/

E-Cell http://www.e-cell.org/

EMA Ecole des Mines d'Alès – http://www.ema.fr

EMBL European Molecular Biology Laboratory – http://www.embl.org/

Ensembl http://www.ensembl.org

Ensembl Multi MartView http://www.ensembl.org/Multi/martview

Ensmart http://www.biomart.org

Entrez http://www.ncbi.nlm.nih.gov/Entrez/

Entrez Gene http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene

EP:GO Expression Profiler http://ep.ebi.ac.uk/EP/GO/

Expasy http://expasy.org/

Fasta http://en.wikipedia.org/wiki/Fasta_format

Fatigo http://www.fatigo.org/

Firebird http://www.firebirdsql.org/

FTP File Transfer Protocol – http://tools.ietf.org/html/rfc959

Full-Malaria http://fullmal.ims.u-tokyo.ac.jp/

G2D Candidate Genes to Inherited Diseases – http://www.bork.embl-heidelberg.de/g2d/

Galen http://www.opengalen.org/

GDB The GDB Human Genome Database – http://www.gdb.org/

GEISHA DNA Array Analysis with Geisha – http://www.pdg.cnb.uam.es/blaschke/cgi-bin/geisha

GEMBio http://www.bioinfo.ensmp.fr/index.php?page_name=ResearchOverview&wikipage=GemBio

GenBank http://www.ncbi.nlm.nih.gov/Genbank/

GenBank http://www.ncbi.nlm.nih.gov/Sitemap/samplerecord.html

Gene Lynx A portal to mammalian genomes – http://www.genelynx.org/

GeneCards http://www.genecards.org

GeneDB http://www.genedb.org/

Genew http://www.gene.ucl.ac.uk/cgi-bin/nomenclature/searchgenes.pl

GenMAPP http://www.genmapp.org/introduction.asp

Genome Express http://www.genome-express.com

Genostar http://www.genostar.com

GEO Gene Expression Omnibus – http://www.ncbi.nlm.nih.gov/geo/

GéoPortail http://www.geoportail.fr/

GeWare Genetic Data Warehouse – https://ducati.izbi.uni-leipzig.de/Geware/servlet/de.izbi.geware.common.forms.FrameSet

GFINDer Genome Function INtegrated Discoverer – http://www.medinfopoli.polimi.it/GFINDer/

GO Gene Ontology – http://www.geneontology.org/

GOA Gene Ontology Annotation Database – http://www.ebi.ac.uk/GOA/

GoMiner http://discover.nci.nih.gov/gominer/

Google Earth http://earth.google.com/

Google Maps http://maps.google.com/

GoSurfer http://www.biostat.harvard.edu/complab/gosurfer/

GOTM Gene Ontology Tree Machine – http://genereg.ornl.gov/gotm

GRAIL The GALEN representation and integration language – http://www.opengalen.org/sources/sources.html

GSDB The Genome Sequence Database – http://www.ncgr.org

GUS The Genomics Unified Schema – http://www.gusdb.org/

HCE Hierarchical Clustering Explorer – http://www.cs.umd.edu/hcil/hce/

HGNC HUGO Gene Nomenclature Committee – http://www.gene.ucl.ac.uk/nomenclature/

Hibernate http://www.hibernate.org/

Hozo http://www.ei.sanken.osaka-u.ac.jp/hozo/eng/index_en.php

HTML HyperText Markup Language – http://www.w3.org/TR/html/

HTTP HyperText Transfer Protocol – http://tools.ietf.org/html/rfc2616

HUGO The Human Genome Organisation – http://www.hugo-international.org/

Hybrigenics http://www.hybrigenics.com/index.html

HypViewer http://graphics.stanford.edu/~munzner/h3/

ICD International Classification of Diseases – http://www.who.int/classifications/icd/en/

IGN Institut Géographique National – http://www.ign.fr/accueil.htm

IMDB International Movie Database – http://www.imdb.com/

INIST L'INstitut de l'Information Scientifique et Technique du CNRS – http://www.inist.fr/

INRA Institut National de la Recherche Agronomique – http://www.inra.fr/

INSDC International Nucleotide Sequence Database Collaboration – http://www.insdc.org/

INSERM Institut National de la Santé et de la Recherche Médicale – http://www.inserm.fr/

Institut Pasteur de Paris http://www.pasteur.fr

InterPro http://www.ebi.ac.uk/interpro/

InXight http://www.inxight.com/

ISO International Standard Organisation – http://www.iso.org

ISoft http://www.isoft.fr

Isys http://sourceforge.net/projects/isys2/

Java http://java.sun.com/

JDBC Java Database Connectivity – http://java.sun.com/javase/technologies/database/

JDBC Java Database Connectivity – http://java.sun.com/javase/technologies/database/

JDO Java Data Objects – *http://java.sun.com/products/jdo/*

Jena *http://jena.sourceforge.net/*

JOBIM Journées Ouverte pour la Biologie, l'Informatique et les Mathématiques
http://crfb.univ-mrs.fr/jobim2007/

JSP Java Server Pages – *http://java.sun.com/products/jsp/*

KAON *http://km.aifb.uni-karlsruhe.de/kaon2/users*

Kartoo *http://kartoo.com/*

KDOM API knowledge discovery object model – *http://www.bcgsc.ca/bioinfo/software*

KEGG Kyoto Encyclopedia of Genes and Genomes – *http://www.genome.jp/kegg/*

KIND *http://www.npaci.edu/DICE/Neuro/*

LabView *http://www.ni.com/labview/*

Latex *http://www.latex-project.org/*

LinkDB/DBGet *http://www.genome.jp/dbget/*

Loom *http://www.isi.edu/isd/LOOM/*

LSID *http://lsid.sourceforge.net*

MAGE MicroArray and Gene Expression – *http://www.mged.org/Workgroups/MAGE/mage.html*

Mappa.Mundi *http://www.mundi.net/*

MatchMiner *http://discover.nci.nih.gov/matchminer/index.jsp*

Matlab *http://www.mathworks.fr/products/matlab/*

MedGene Database *http://hipseq.med.harvard.edu/MEDGENE/*

MedMiner *http://discover.nci.nih.gov/textmining/main.jsp*

MedMole Mining On-Line Expert on MedLine – *http://medmole.cineca.it/*

MeKE (Medical Knowledge Explorer) – *http://gen.csie.ncku.edu.tw/meke3/*

MeSH Medical Subject Headings – *http://www.nlm.nih.gov/mesh/*

MeSH Bilingue *http://ist.inserm.fr/basismesh/mesh.html*

MetaFam *http://metafam.ahc.umn.edu/*

MeV MultiExperiment Viewer *http://www.tm4.org/mev.html*

MGED Society Microarray Gene Expression Data Society – *http://www.mged.org/*

MGI Go Browser (Mouse Genome Informatics) – *http://www.informatics.jax.org/searches/GO_form.shtml*

MIAME Minimum information about a microarray experiment –
http://www.mged.org/Workgroups/MIAME/miame.html

Mix *http://db.ucsd.edu/Projects/MIX/*

MKBeem *http://www.mkbeem.com/*

MPIM Mitochondrial Protein Import Machinery –
http://www.plantenergy.uwa.edu.au/applications/mpimp/index.html

MRNT Ministère de la Recherche et des Nouvelles Technologies – *http://www.recherche.gouv.fr/*

myGrid *http://www.mygrid.org.uk/*

MySQL *http://www.mysql.com/*

NAR Nucléic Acids Research – *http://nar.oxfordjournals.org/*

NCBI National Center for Biotechnology Information – *http://www.ncbi.nlm.nih.gov/*

NCBI Taxonomy *http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Taxonomy*

NIH National Institute of Health – *http://www.nih.gov/*

NLM National Library of Medicine – *http://nar.oxfordjournals.org/*

NRCC National Research Council of Canada – *http://www.nrc-cnrc.gc.ca*

ODBC Open DataBase Connectivity – *http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/htm/dasdkodbcoverview.asp*

ODF OASIS Open Document Format – http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1-html/OpenDocument-v1.1.html

ODL http://www.odmg.org/wrayjohnson.htm

ODMG Object Data Management Group – http://www.odmg.org/

OIL http://www.w3.org/TR/daml+oil-reference

OilEd http://oiled.man.ac.uk/

OLAP On Line Analytical Processing –

OMG Object Managment Group – http://www.omg.org/

OMIM Online Mendelian Inheritance in Man – http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=OMIM

OntoEdit http://www.ontoprise.de/products/ontoedit_en

Ontolingua http://www.ksl.stanford.edu/software/ontolingua/

OntoSaurus http://www.isi.edu/isd/ontosaurus.html

OntoTools http://vortex.cs.wayne.edu/projects.htm

OOA http://www.sei.cmu.edu/str/descriptions/ooanalysis_body.html

Open XML Microsoft Office 2007 http://www.microsoft.com/france/msdn/office/Presentation-formats-de-fichier-Open-XML.mspx#E4C

OQL Object Query Language, Site de l'ODMG (Object Data Management Group) – http://www.odmg.org/

Oracle http://www.oracle.com/

OWL Web Ontology Language – http://www.w3.org/2004/OWL/

Panther Protein ANalysis THrough Evolutionary Relationships – http://www.pantherdb.org/

Parvis http://home.subnet.at/flo/mv/parvis/

PDB Protein Data Bank – http://www.rcsb.org/pdb/home/home.do

Pétillant http://www.petillant.com/

Picsel Production d'Interfaces à base de Connaissances pour des Services En Ligne – http://www.lri.fr/LRI/iasi/picsel/

PIR Protein Information Resource – http://pir.georgetown.edu/

PIR-PSD PIR Protein Sequence Database – http://pir.georgetown.edu/pirwww/dbinfo/pir_psd.shtml

PlasmoDB The Malaria Parasite Genome Resource – http://www.plasmodb.org/plasmo/home.jsp

PMC PubMed Central – http://www.pubmedcentral.nih.gov/

PostGres SQL www.postgresql.org/

ProActive http://www-sop.inria.fr/oasis/ProActive/

Protégé2000 http://protege.stanford.edu/

PSU Pathogen Sequencing Unit – http://www.sanger.ac.uk/Projects/Pathogens/

PubGene PubGene Gene Database and Tools – http://www.pubgene.org/

PubMed http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?DB=pubmed

QuickGO http://www.ebi.ac.uk/ego/

R The R Project – http://www.r-project.org/

RCSB Research Collaboratory for Structural Bioinformatics – http://home.rcsb.org/

RDF Ressource Description Framework – http://www.w3.org/RDF/

Reactome http://www.reactome.org/

Refseq NCBI Reference Sequences – http://www.ncbi.nlm.nih.gov/RefSeq/

RMI Remote Method Invocation – http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp

SciLab http://www.scilab.org/

SeqHound http://www.blueprint.org/seqhound/

SGML Standard Generalized Markup Language – *http://www.w3.org/MarkUp/SGML/*

SIB Swiss Institute of Bioinformatics – *http://www.isb-sib.ch/*

Smartmoney *http://www.smartmoney.com/marketmap/*

SMIL Synchronized Multimedia Integration Language (SMIL 3.0 - draft) – *http://www.w3.org/TR/SMIL3/*

S-Moby *http://semanticmoby.org/*

SnomedCT the Systematized Nomenclature of Medicine – Clinical terms – *http://www.snomed.org/snomedct/index.html*

SOAP Simple Object Access Protocol – *http://www.w3.org/2002/07/soap-translation/soap12-part0.html*

Source *http://source.stanford.edu/cgi-bin/source/sourceSearch*

SPARQL *http://www.w3.org/TR/rdf-sparql-query/*

Specialist Lexicon (NIH – UMLS) *http://lexsrv3.nlm.nih.gov/SPECIALIST/Projects/lexicon/current/index.html*

Specialist NLP Tools (NIH – UMLS) *http://lexsrv3.nlm.nih.gov/SPECIALIST/index.html*

SRS *http://srs.ebi.ac.uk/*

SwissProt *http://expasy.org/sprot/*

SYGMART *http://www.lirmm.fr/~chauche/PresentationSygmart.html*

Syntex *http://www.irit.fr/RFIEC/syntex/*

TAIR The Arabidopsis Information Resource – *http://www.arabidopsis.org/*

Taverna *http://taverna.sourceforge.net/*

TermSciences *http://www.termsciences.fr/article.php3?id_article=40*

TIGR The Institute for Genomic Research – *http://www.tigr.org/*

TouchGraph A. Shapiro – *http://www.touchgraph.com*

ToxNuc-E *http://www.toxnuc-e.org/*

Tree Tagger *http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html* , testable en ligne : *http://cental.fltr.ucl.ac.be/~pat/tagger/*

TrEMBL *http://www.ebi.ac.uk/trembl/index.html*

TXTGate *http://tomcat.esat.kuleuven.be:8080/txtgate/home.jsp*

UCGIS University Consortium for Geographic Information Science – *http://www.ucgis.org/*

UCSC Genome Browser *http://genome.ucsc.edu/*

UML Unified Modeling Language – *http://www.uml.org/*

UMLS Unified Medical Language System – *http://umlsinfo.nlm.nih.gov/*

UMLSKS *http://umlsks.nlm.nih.gov/kss/servlet/Turbine/template/admin,user,KSS_login.vm*

UniGene *http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=unigene*

UniParc *http://www.ebi.uniprot.org/uniprot-srv/uniParcSearch.do*

Uniprot The Universal Protein Resource – *http://www.expasy.uniprot.org/*

UniProtKB The Uniprot Knowledge Base – *http://www.expasy.uniprot.org/database/knowledgebase.shtml*

UniRef *http://www.ebi.uniprot.org/uniprot-srv/uniRefSearch.do*

URI/URL Universal Resource Identifier/Locator– *http://www.w3.org/Addressing/*

V-Cell *http://www.vcell.org*

W3C World Wide Web Consortium – *http://www.w3.org/*

WebODE *http://www-sop.inria.fr/acacia/ekaw2000/ode.html*

WebOnto *http://kmi.open.ac.uk/projects/webonto/*

WebSphere Information Integrator *http://www-03.ibm.com/industries/healthcare/doc/content/solution/939513105.html*

Wikipedia *http://www.wikipedia.org/*

WikiViz *http://www.wikiviz.org*

WIRM Web Interface Repository Manager – *http://www.wirm.org/*

Wormbase *http://www.wormbase.org/*

WTSI Wellcome Trust Sanger Institute – *http://www.sanger.ac.uk/*

XML Extensible Markup Language – *http://www.w3.org/TR/xml11/*

XPath *http://www.w3.org/TR/xpath*

XPointer *http://www.w3.org/TR/WD-xptr*

XQuery *http://www.w3.org/TR/xquery/*

XSLT Extended Stylesheet Language Transformations – *http://www.w3.org/TR/xml11/*

Partie 3

Annexes

Annexe A Notions complémentaires sur les systèmes d'information et l'ingénierie des connaissances

A.1 Eléments généraux de génie logiciel

A.1.1 Architecture logicielle

> *Tous les modèles architecturaux de référence applicables aux systèmes interactifs répondent au requis de modifiabilité : en l'état des connaissances et des pratiques, la mise au point itérative d'une IHM*[1] *est la seule solution effective. Mais le succès de cette approche repose soit sur l'existence de bons outils de construction de maquettes, soit sur la possibilité de retoucher « à la main » une IHM sans mettre en cause la fiabilité, ni atteindre des coûts prohibitifs de mise à jour. [Coutaz and Nigay 2001]*

Cette réflexion peut s'élargir à la conception de logiciels et même plus généralement à l'ingénierie. Les produits logiciels évoluent rapidement ; ils sont de plus en plus complexes et doivent être de plus en plus interopérables. Les composantes d'un logiciel doivent être capables d'évoluer sans remettre en cause la fiabilité et la sécurité de l'intégralité du système. Joëlle Coutaz et Laurence Nigay évoquent le *principe de séparation fonctionnelle* et définissent ainsi l'architecture :

> *L'architecture d'un système informatique est un ensemble de structures comprenant chacune : des composants, les propriétés extérieures visibles de ces composants et les relations qu'ils entretiennent. (Définition inspirée de [Bass, Faneuf et al. 1992] dans [Coutaz and Nigay 2001]).*

La réutilisabilité et l'autonomie de composantes est la motivation de plusieurs paradigmes de la programmation : objets, agents, composants, services, etc. A un plus haut niveau, ce *principe de séparation fonctionnelle* est étendu par les modèles de référence utilisés dans la conception d'applications interactives (Seeheim [Pfaff 1985], Arch [Bass, Faneuf et al. 1992], MVC [Krasner and Pope 1988], PAC [Coutaz 1990], PAC-Amodeus [Nigay 1994], *confer* [Coutaz and Nigay 2001] ou [Patry 1999] pour une revue détaillée). Ce principe est appliqué dans le contexte de la publication de documents, où les systèmes et standards évoluent de manière à séparer le contenu de sa mise en forme (SGML & DSSSL, HTML & CSS, XML & XSLT, les feuilles de styles de LaTeX et des traitements de textes).

La notion de client et de serveur repose sur le besoin de partage d'une ressource entre plusieurs utilisateurs. Le **serveur** est le logiciel qui contrôle la ressource, assure le partage, garantit la sécurité, l'intégrité, etc. Un **client** est un logiciel qui expédie des requêtes au serveur et en attend la réponse. La notion d'architecture **n-tiers** consiste en la séparation fonctionnelle sur plusieurs serveurs. Un **intergiciel** (*middleware*) est un outil intermédiaire entre le niveau

[1] Interface homme-machine

client et serveur qui facilite la communication et l'interopérabilité entre plusieurs applications ou niveau d'applications. Il a pour rôle d'assurer des services (gestions de l'envoi de messages, service d'abonnements, transmission de flux de données, distribution sur un réseau, asynchronisme, erreurs, appel distant de procédures, etc.) et d'ajouter une indépendance. Par exemple, JDBC est une librairie permettant d'interroger un SGBD[1] *via* une interface de programmation (API) orientée objet. Un pilote assure la connexion au SGBD. D'un point de vue théorique, cela permet de changer de SGBD sans avoir à modifier le code de l'application.

Alors que la séparation fonctionnelle améliore la modifiabilité d'une application, la centralisation (par un serveur ou toute autre approche) contribue à la souplesse de déploiement et à la modifiabilité de la donnée. Si on centralise le code d'une application, l'évolution du code implique la mise à jour du système central et évite les coûts de déploiement des mises à jour sur tous les systèmes clients. La centralisation de données permet d'éviter la redondance. Dès lors qu'il y a redondance, la mise à jour d'une donnée doit être répercutée sur toutes les copies pour maintenir l'intégrité du système. La centralisation réduit le coût des mises à jour du code applicatif ou des données. L'illustration de ces deux concepts est parfaitement incarnée par la recrudescence de logiciels en ligne. Le client graphique spécifique disparait au profit du navigateur livré avec tous les systèmes d'exploitation modernes. La gestion de l'affichage et des données est assurée par le serveur. La modification d'une donnée unique est répercutée directement sur la communauté d'utilisateurs, et la centralisation du code sur le serveur évite les coûts de déploiement et de mises à jour sur les postes clients. Au sein du serveur même, la gestion des données et des opérations métiers (noyau fonctionnel) et l'interface utilisateur obéissent au principe de séparation fonctionnelle. C'est dans cette optique d'indépendance de l'application par rapport au schéma que l'on met en œuvre des vues dans une base de données (cf. section A.1.2.4 page 263).

A.1.2 Systèmes d'information

Nous proposons une définition plus large de la base de données fondée sur les ouvrages de Serge Miranda d'une part [Miranda 2002] et Georges Gardarin d'autre part [Gardarin 1999; Miranda 2002]. Les notions clés associées que nous employons dans la suite de ce mémoire sont présentées dans cette section : le système de gestion de bases de données (SGBD), ses fonctionnalités essentielles, le modèle, le schéma, les métadonnées, les vues et leur matérialisation, etc.

A.1.2.1 Base de données, modèle et schéma

Une définition générale de la base de données est un regroupement de données persistantes structurées par un schéma. Ce schéma peut être spécifié dans différents paradigmes (modèle relationnel, modèle objet, etc.). Suivant les principes architecturaux précédents, on recommande généralement une séparation entre le schéma interne des données et la vue qu'en a un groupe d'utilisateurs. Ces notions sont développées dans cette section. A partir de la définition de *« donnée »* de l'AFNOR, Serge Miranda définit la base de données en regard de celle d'un fichier :

[1] Les bases de données et systèmes de gestion de bases de données (SGBD) sont présentés dans la section suivante).

> *Un fichier est un ensemble de données informatiques qui peut être manipulé par plusieurs utilisateurs ayant une vue unique de ces données. Des vues multiples peuvent être obtenues par tri des données stockées. Un fichier est un ensemble d'enregistrements physiques (ou « articles ») eux même composés de « champs ». […]* ***Une base de données est un ensemble de données*** *qui peut être manipulé par plusieurs utilisateurs****, ayant des vues différentes sur ces données.*** *Une base de données est le regroupement d'un ensemble de fichiers partagé par des utilisateurs différents, concurrents, et… compétiteurs.* ***La structure de cet ensemble requiert une description rigoureuse que l'on appellera « schéma ».*** *[Miranda 2002]*

Les différents fichiers utilisés pour stocker les données au niveau physique ne sont pas indépendants, ils sont structurés par un schéma. Le modèle permet de spécifier ce schéma et les contraintes qui y sont appliquées. Il fournit de plus des mécanismes d'opérations adaptés aux problématiques de la gestion des données. Serge Miranda définit le modèle comme suit :

> *Un modèle de donnée est un ensemble de structures et un ensemble d'opérations définies dessus et des mécanismes de contrôle associés correspondant au triptyque Définition – Manipulation – Contrôle. [Miranda 2002]*

Le modèle permet de représenter (modéliser) un domaine, un problème, et met à disposition un ensemble d'opérations et d'algorithmes adaptés à la manipulation des données (création, suppression, modification, sélection, etc.). Le fruit de cette modélisation est le schéma. Le modèle défini par S. Miranda possède une définition proche de la notion de paradigme. Le terme modèle est aussi fréquemment employé comme le résultat de la modélisation, c'est-à-dire le schéma. Afin de lever toute ambiguïté, dans la suite de ce document, nous désignerons le modèle défini par Miranda par le terme « paradigme ». Le terme « modèle » sera employé comme synonyme de schéma et défini comme suit :

> *Nous appelons schéma de données l'abstraction résultant de l'application d'un modèle de données à une entreprise. […] Nous appelons base de données l'ensemble de données informatiques associées à un schéma de données et physiquement stockées en mémoire. [Miranda 2002]*

Plusieurs modèles existent ; le modèle relationnel est ancien et le plus répandu. Introduit par Edgar Codd en 1970, ce modèle repose sur l'algèbre relationnelle, elle-même fondée sur la théorie mathématique des ensembles [Codd 1969; Codd 1970]. Les données sont stockées dans des « tables » appelées **relations** (des ensembles). Une « colonne » de la table est un **attribut** de la relation. Cet attribut est défini sur un **domaine** : l'ensemble de valeurs que peut prendre l'attribut. Ceci est différent du type : par exemple, l'année de naissance d'une personne est un entier (type – niveau syntaxique) positif inférieur ou égal à l'année courante (domaine – sémantique). Les éléments de l'ensemble contenus dans les relations sont les **tuples**. Pour concevoir un schéma relationnel, il existe plusieurs méthodologies : modèle entités-relations [Chen 1976], Merise [Tardieu, Rochfeld et al. 1983; Tardieu, Rochfeld et al. 1985; Rochfeld and Morejon 1989], NIAM et ORM [Nijssen and Halpin 1989; Halpin 2001]. Une bonne pratique pour concevoir (ou vérifier) un schéma est de rechercher les dépendances fonctionnelles. Cette approche est dite *par synthèse* [Gardarin 1999]. A partir de la liste des attributs et de leurs dépendances, on génère un schéma complet. Une dépendance fonctionnelle est définie comme suit :

Soit $\mathcal{R}(A_1, A_2, \dots, A_n)$ un schéma de relation, et X et Y deux sous-ensembles de $\{A_1, A_2, \dots, A_n\}$. On dit que X détermine Y (ou que Y dépend fonctionnellement de X) si, et seulement si, des valeurs identiques de X impliquent des valeurs identiques de Y. On le note $X \rightarrow Y$. Autrement dit, $X \rightarrow Y$ si et seulement si :

$$\text{Pour tous tuples } t_1, t_2 \in \mathcal{R}: t_1[X] = t_2[X] \rightarrow t_1[Y] = t_2[Y]$$

Cette approche formelle permet, entre autres, la définition de formes normales [Codd 1970]. Il s'agit d'un ensemble de règles qui permet, si elles sont respectées dans un schéma donné, de garantir l'absence de redondances.

Des alternatives au modèle relationnel existent. Les pionniers sont le modèle réseau [Codasyl 1971] et le modèle hiérarchique [Tsichritzis and Lochovsky 1976; Gouy, Milleret et al. 1984], regroupés sous le terme de modèle navigationnel. Ces approches appelées *légataires* par Georges Gardarin [Gardarin 1999] inspirent toujours des travaux concernant les systèmes de fichiers, bases de registre ou encore certaines bases de données XML natives. Les approches logiques toutes aussi anciennes ont abouti vers la fin des années 70 à des bases de données déductives [Gallaire and Minker 1978] : on stocke un ensemble de règles et de données factuelles, des mécanismes d'inférence sont adjoints à la manipulation des données. Cette approche est toujours exploitée par la communauté intelligence artificielle et ingénierie des connaissances dans le contexte des bases de connaissances, systèmes experts, et bases de données RDF, par exemple. Peu de temps après le modèle relationnel, le paradigme objet a fait son apparition. Il a introduit la notion d'encapsulation et d'héritage. L'approche objet-relationnel réconcilie ces deux paradigmes principaux. Il apporte entre autres des algorithmes de traduction de schémas d'un paradigme vers l'autre, et ajoute l'encapsulation et l'héritage au relationnel. Cette approche est notamment fréquemment employée par les intergiciels de persistance d'objets. [Gardarin 1999] et [Miranda 2002] proposent une revue complète de l'objet et du relationnel.

A.1.2.2 Système de gestion de bases de données (SGBD)

La base de données est une collection de données organisée à l'aide d'un schéma qui lui-même doit respecter certaines règles et contraintes (forme normales, modèle représentant un domaine, etc.). Le SGBD permet de manipuler ces données. Cette définition est assez large, et la communauté considère qu'un *véritable* SGBD se doit de posséder un minimum de fonctionnalités, notamment la gestion des transactions. E.F. Codd considère ainsi 12 règles indispensables [Codd 1985; Codd 1985]. On cite aussi fréquemment les propriétés *ACID (Atomicity, Consistency, Isolation, Durability)*[Gray 1981] garantissant l'intégrité des données, la fiabilité du système et sa tolérance aux pannes (mécanisme de transactions).

Notons qu'une fonctionnalité essentielle du SGBD est la mise à disposition de langage de requête (SQL en relationnel [Chamberlin and Boyce 1974], OQL en orienté objet, Datalog en déductif et XQuery, XPath et XPointer en XML, SPARQL pour RDF). Ces langages sont non procéduraux et permettent de spécifier le résultat que l'on souhaite sans décrire comment calculer ce résultat. La planification de la requête est alors déléguée au SGBD. Cela n'interdit cependant pas la présence dans les SGBD principaux de langages procéduraux complémentaires [Feuerstein and Pribyl 2005].

Dans la pratique, tous les SGBD n'implémentent pas parfaitement toutes ces fonctionnalités et n'en respectent pas complètement les standards. De la même façon qu'il existe différents paradigmes, il existe différents types de SGBD : navigationnels (hiérarchiques et réseaux), déductifs, relationnels, objet, objet-relationnels, XML, temporels, multidimensionnels, etc. MySQL est un système très controversé. Très longtemps, il n'était pas considéré comme un « bon » SGBD car il ne gérait par les transactions. Actuellement il a fait de gros progrès, mais n'équivaut toujours pas des systèmes concurrents comme PostGres SQL ou Oracle. Malgré ce, il est très souvent utilisé pour des questions de performances, de simplicité de mise en œuvre, ou tout simplement sa présence historique dans la communauté. Nous ne développerons pas plus l'inventaire des systèmes existants.

Banque de données et système documentaire

On rencontre également le terme « banque de données », particulièrement dans le contexte de la bioinformatique. Une définition est proposée par S. Miranda :

> *Une base de données sous entend un type de données « factuelles » ou (« primaires ») alors qu'une banque de données, un type de données « référentielles » (ou « secondaires »).*

Concrètement, une base de données est le stockage structuré de données organisées par un schéma. La première forme normale de Codd précise que les attributs de ces schémas doivent

être atomiques[1]. Dans une banque de données, on structure non pas des données, mais des documents ou fichiers. La base contient des références (pointeurs, liens) vers les fichiers qui ne sont pas obligatoirement structurés. A l'apparition des premiers projets bioinformatiques liés au séquençage, on entreposait ainsi chaque séquence sous forme d'un fichier dans un format de type Fasta ou GenBank, par exemple (cf. annexe C.1 page 275). C'est la naissance des banques de données génomiques. Pour Miranda, les systèmes documentaires proposent des services comparables :

> *Les systèmes documentaires ont pour fonction première d'offrir comme information une indirection sur un texte (livre, article…) contenant le résultat recherché, alors qu'un SGBD fournit directement ce résultat.*

Les systèmes documentaires sont des outils de recherche de documents, qui, souvent, ne possèdent pas le contenu des documents, mais uniquement des données bibliographiques (métadonnées, cf. section suivante) et des liens d'accès (bibliothèques, librairies, éditeurs, etc.). Un distinction entre banque de données et système documentaire, outre la restriction du domaine bibliographique, est qu'un système documentaire ne stocke pas forcément les documents localement, alors que la banque de donnée génomique stocke ses fichiers de séquences.

A.1.2.3 Metadonnées

Etymologiquement, le préfixe *meta* provient du grec où il signifie *« après »*. Les livres de « la physique de la physique » étant classés après ceux de la physique sur l'étagère d'Aristote, cette discipline a ainsi été nommée métaphysique. Actuellement, le préfixe *meta* signifie un degré de *réflexion*[2] : Les métadonnées sont *des données sur les données*. Dans le contexte d'une base de données, les métadonnées s'articulent généralement autour de deux éléments : la description des schémas et structures accessibles au sein du SGBD (niveau syntaxique), les structurations sémantiques liées à la base de données (mais pas obligatoirement au SGBD).

Les principaux SGBD actuels proposent un accès à des métadonnées. Dans ces métadonnées, les schémas sont des données. Dans le cas d'un modèle relationnel, il y a une table qui contient la liste des tables, une autre celle des attributs, encore une autres celles des contraintes. Les dépendances entre les attributs et les tables sont elles mêmes représentées par des contraintes sur le métamodèle.

Alors que les métadonnées partagées par le SGBD se situent à un niveau syntaxique, d'autres représentent un support sémantique. Des métamodèles sont parfois employés comme support sémantique du schéma. Les terminologies (en particulier les vocabulaires contrôlés) sont utilisées pour structurer sémantiquement les valeurs prises par les données. C'est le rôle de MeSH pour PubMed et de Gene Ontology pour l'annotation des gènes et de leurs produits. Ces deux types de supports permettent l'inférence à partir des données et leur analyse statistique et probabiliste.

A.1.2.4 Séparation fonctionnelle : Vue & matérialisation

[1] Par exemple, le numéro de sécurité social devrait être décomposé champs numériques et non stocké comme une chaîne de caractère unique. De même, on sépare noms et prénoms d'une personne, ou encore les différentes composantes de son adresse (numéro, rue, code postal, ville, etc.).

[2] au sens de renvoi sur soi et non d'acte de pensée

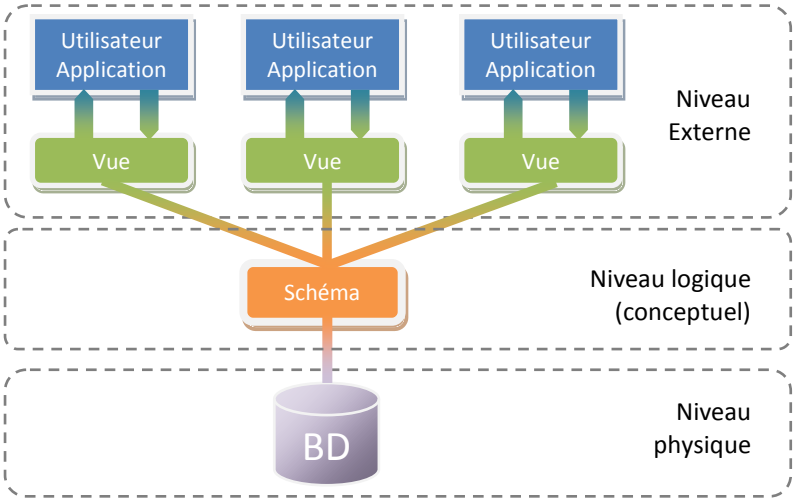


Figure A.1 – L'utilisateur ne manipule pas directement le schéma mais une vue externe sur ce schéma. Cette vue est une représentation adaptée à la tâche de l'utilisateur alors que le schéma interne et lié à l'expression de contraintes, l'optimisation de calculs, etc. qui ne le rendent pas toujours suffisamment intuitif pour l'utilisateur.

Nous avons présenté dans la section précédente quelques notions élémentaires de génie logiciel et plus particulièrement d'architecture. Une des principales recommandations en la matière est de séparer le noyau fonctionnel de la vue externe. Ce principe est appliqué au sein des SGBD : le schéma est isolé du niveau physique d'organisation des données (fichiers). De même, lorsque le schéma est construit, l'utilisateur n'y a accède pas directement mais par l'intermédiaires de vues et de procédures stockées [Baril 2003].

Le niveau logique est aussi appelé parfois niveau conceptuel. Il décrit l'organisation logique des données, sans se préoccuper de leur stockage physique. Le niveau externe permet de spécifier l'interface de la base de données avec les applications ou les utilisateurs. Un mécanisme de vue permet de restructurer les données dans un SGBD. G. Gardarin définie la vue comme suit :

> *Une ou plusieurs tables virtuelles dont le schéma et le contenu sont dérivés de la base réelle par un ensemble de questions.*

Les vues peuvent être utilisées dans plusieurs buts : limiter l'accès à un sous ensemble des données ou du schéma (pour le confort, la sécurité ou la confidentialité par exemple), simplifier l'utilisation de la base de données en unifiant des entités, etc. Dans le contexte relationnel, la définition de la vue est une requête SQL.

Le schéma relationnel (interne) est conçu pour modéliser un domaine en respectant certaines règles (formes normales, etc.). Ce schéma est souvent complexe et peu intuitif. Il est aussi amené à évoluer. Les vues permettent de définir un schéma virtuel vu par l'utilisateur et l'application cliente. Cette séparation entre le schéma interne et le schéma externe apporte une indépendance : si le schéma interne évolue, il n'est pas nécessaire de modifier l'application cliente ; l'évolution est transparente au niveau externe. Réciproquement, on peut faire évoluer le client sans remettre en cause l'intégrité, la fiabilité et la sécurité au niveau interne. Les vues permettent aussi de simplifier la manipulation des données par l'utilisateur, d'améliorer la sécurité, la confidentialité. Les vues permettent d'adopter un point de vue différent sur le schéma, mais aussi sur les données : la vue est une requête SQL, elle inclue donc potentiellement des fonctions de calcul, des critères de tris, des filtres de sélection, etc. En complément des vues (accès aux données) il est nécessaire de retrouver une séparation comparable pour la modification des données. C'est l'un des rôles des procédures stockées.

Les vues « virtuelles » ne stockent pas leurs données, elles sont remises à jour à chaque invocation. Ceci implique parfois un coût de calcul important. Il est parfois intéressant de stocker les données contenues dans les vues. On parle alors de vues « matérialisées ». L'approche matérialisée induit la redondance des données. Le choix entre approche matérialisée

et approche virtuelle est motivé par un rapport entre, d'une part le coût et la fréquence de calcul des requêtes, d'autre part le coût et la fréquence des mises à jour. Les vues matérialisées sont parfois employées uniquement motivées par des économies de calculs dans certains systèmes.

A.2 Exemples de systèmes d'intégration

A.2.1 Entrepôt

Les besoins en termes de propriété et de confidentialité des données ont motivé de nombreuses initiatives de conception d'entrepôts de données. La suite présente des systèmes appliqués aux données biologiques. Les expériences reposent majoritairement sur le paradigme relationnel, le reste reposant sur le paradigme objet. Les seules autres approches dont nous ayons connaissances sont **Acnuc** [Gouy, Milleret et al. 1984], l'un des premiers systèmes, qui repose sur un modèle réseau, et **GeWare**. Ce dernier se focalise sur les données d'expression et adopte un modèle multidimensionnel [Rahm, Kirsten et al. 2007]. Enfin, **WIRM** (Web Interface Repository Manager) se définit plus comme un environnement de développement rapide d'application bioinformatiques possédant des données intégrées et matérialisées (entrepôts). Il s'appuie sur des serveurs relationnels.

Approches relationnelles

GUS (*Genomics Unified Schema*) - Dans le contexte de la biologie, l'un des projets les plus avancé est l'entrepôt *GUS* [Davidson, Crabtree et al. 2001]. Il est actuellement mis en œuvre dans différents contextes dont *GeneDB* (projets de séquençage du *Trust Sanger Institute)* et *PlasmoDB (Plasmodium Falciparum)*. Il existe de l'ordre d'une vingtaine de projets utilisant GUS. GUS intègre des données sur les séquences, leur cartographie, les domaines des protéines, les données d'expression au format MAGE/MIAME, et de nombreuses ontologies. De nombreux outils sont aussi inclus pour la prédiction de gènes et la recherche de séquence similaires. L'intégration y est syntaxique : le schéma est l'union des sources, découpé en 5 domaines. *DoTS (**D**B **o**f **T**ranscribed **S**eqs)* contient toutes les informations sur les gènes, leurs séquences, les transcrits et les protéines et leurs annotations. *SRes (**S**hared **r**esources)* contient des ontologies, taxonomies, etc. *Rad (**R**NA **A**bundance **D**B)* se focalise sur les données d'expression, et *TESS (**T**rans **E**lem **S**earch **S**ite)* sur la régulation des gènes. Chaque domaine est un projet réutilisable indépendamment, et historiquement étranger à GUS. Le dernier domaine, appelé *Core*, est un ensemble de données permettant de conserver une traçabilité sur les algorithmes et leur mise en œuvre. GUS est en effet reconnu pour ses qualités de traçabilité de l'information et de leur transformation. GUS repose sur le paradigme relationnel avec une surcouche objet fournie par Oracle. Bien que l'intégration soit syntaxique, un ensemble d'algorithmes d'intégration verticale sont mis en œuvre et leurs résultats sont stockés dans une partie de l'entrepôt. De plus, les schémas des différents projets ne sont pas liés à une source unique mais des modèles de certaines problématiques. Pour ces raisons, on considère souvent GUS comme un entrepôt d'intégration intermédiaire entre le sémantique et le syntaxique. Notons enfin qu'il est fourni avec un portail en ligne proposant de nombreux services (un exemple de page est proposé dans l'annexe C.1.1 (294).

Atlas – Atlas est un entrepôt relationnel qui intègre des données depuis de nombreuses ressources sur les gènes, les protéines et leurs interactions, ainsi que des ontologies [Shah, Huang et al. 2005]. L'intégration y est sémantique ; Elle repose sur un schéma global modélisateur. Aucune information n'est donnée sur l'intégration verticale. La mise à jour des données se fait par un rechargement complet des données, mis à part pour les données de GenBank qui disposent d'une procédure incrémentale. L'intégration se fait au travers de différentes technologies : C/C++, Java, SQL et Perl. Il propose des accès aux données en SQL ou via une API Java.

BioWarehouse – BioWareHouse est un entrepôt relationnel (Oracle ou MySQL) libre, orienté vers la représentation des voies biologiques (métaboliques, de signalisation, etc.) [Lee, Pouliot et al. 2006]. Il est interrogeable en SQL et OOA. Une première particularité de l'approche est la volonté de réduction de la taille du schéma en mutualisant certaines relations. Par exemple, une seule table contient les commentaires, qu'il s'agisse d'un gène, d'une protéine ou d'un transcrit. Malgré cela, le schéma possède tout de même 179 relations dont 120 sont des entités. Les procédures d'intégration sont, ici aussi, hétérogènes : C et Java.

EnsMart – Ensmart est un entrepôt relationnel qui intègre les données d'Ensembl et d'autres bases de données reliées [Kasprzyk, Keefe et al. 2004]. L'intégration se déroule en deux étapes. La première récupère les données externes et les charge dans le SGBDR en utilisant les schémas locaux. La seconde intègre dans le schéma global chaque base de données intermédiaire en utilisant des procédures spécifiquement développées (en Perl). Les données sont interrogeables en SQL. Une interface en ligne (MartView) est mise à disposition avec une API (Mart API), tout ceci implémenté en Perl. MartJ est une suite implémentée en Java contenant une API (MartLib) et deux applications clientes : MartExplorer qui propose les mêmes fonctionnalités que MartView encapsulées dans une interface graphique, et MartShell, une interface en ligne de commande permettant l'exécution de scripts grâce à un langage dédié.

SeqHound – SeqHound est un entrepôt relationnel construit pour stocker les ressources du NCBI [Michalickova, Bader et al. 2002]. Les auteurs évoquent les limites abordées précédemment liées au portail d'Entrez : problème de performance pour la fouille, de traçabilité et propriété des données, d'accès via un langage de requête évolué, etc. SeqHound est donc un entrepôt qui contient les données de nombreuses bases du NCBI, interrogeable en SQL et fourni avec une API C, C++, Java et Perl fournissant des outils comparables aux services fournis par Entrez. Un avantage de cet entrepôt est la mise à jour quotidienne des données. SeqHound est classé comme système d'intégration grâce à la possibilité d'interroger les données en SQL. L'intégration se fait par la présence d'une relation pour chaque source. Chaque tuple correspond alors à l'équivalent d'un fichier au format ASN.1 ou GenBank. Il est cependant difficile de parler à proprement parler d'intégration de données : les fichiers ne sont pas décomposés en attributs atomiques et entités diverses. Les requêtes SQL permettent de récupérer directement des contenu en format ASN.1 ou GenBank. Ces contenus ne sont pas décomposés en formats atomiques, comme le présuppose la première forme normale de Codd. Il est donc impossible d'interroger les données de façon structurée sur la totalité de leur contenu, cela se fait essentiellement au travers d'identifiants et des références croisées qui sont intégrées. Ce système s'apparente donc à un dépôt de données évolué.

UMLS – Nous avons déjà décrit UMLS dans la section 2.3.1.2 (page 55) [Bodenreider 2004]. Les données sont décomposées en attributs atomiques dans un schéma relationnel et interrogeables en SQL. Un portail en ligne est proposé (UMLSKS) et une application locale permet d'interroger les fichiers sans SGBDR. L'intégration sémantique est horizontale (présence d'un schéma global) et verticale (Metathesaurus qui sépare concept et atomes). Notons que le schéma relationnel ne respecte pas la seconde forme normale de Codd.

Il existe enfin de nombreuses initiatives dont voici une énumération rapide. **Dwarf** repose sur le DGBDR Firebird [Fischer, Thai et al. 2006]. L'interface en ligne est implémentée en Perl. Il se focalise sur les séquences, structures et annotations des protéines. **MetaFam** s'intéresse aux familles de protéines [Silverstein, Shoop et al. 2001]. Il propose notamment un client graphique développé en Java assez évolué. **DiscoveryDB** est l'entrepôt de la plateforme DiscoverySpace [Robertson, Oveisi-Fordorei et al. 2007]. Il intègre 26 sources différentes. Basé sur MySQL, il est actuellement en cours de migration vers PostGres SQL. Sa première originalité est d'exploiter un modèle ontologique, initialement implémenté à l'aide de la KDOM API (a Knowledge Discovery Object Model API) [Zuyderduyn and Jones 2003] et actuellement en cours de migration vers l'API Jena, un intergiciel en Java assurant la persistance pour RDF et OWL [McBride 2001]. Une seconde originalité est l'exploitation du standard de l'OMG et de l'I3C : Life Science Identifier (LSID). **UCSC Genome Browser** repose aussi sur MySQL et se restreint à des données de séquences [Karolchik, Baertsch et al. 2003]. Enfin **CoryneRegNet** se spécialise sur les réseaux de

régulation et les facteurs de transcription du genre de champignon Coryne [Baumbach, Brinkrolf et al. 2006]. Il est développé en Perl pour le portail, en Java pour la visualisation et l'intégration et il utilise MySQL. Il repose en particulier sur un modèle de graphe assez simple et utilise une visualisation avancée basée sur la bibliothèque yFiles. **BioMolQuest** se focalise sur les protéines [Bukhman and Skolnick 2001]. Son originalité provient d'un outil de recherche par mot clé sur la totalité du contenu. Il exploite par ailleurs les références croisées. Il existe de multiples autres entrepôts, plus spécifiques et non publiés. Nous ne pouvons en faire un inventaire exhaustif. De plus, certains entrepôts et certains médiateurs dans leur approche sont parfois considérés comme des plateformes, et sont ainsi présentés dans la section dédiée. C'est le cas d'Isymod et GenoStar par exemple qui se positionnent comme des entrepôts de connaissances utilisés dans une plateforme.

Approches orientées objet

Concernant le paradigme objet, deux entrepôts ont été développés et sont fréquemment référencés : Gedaw et GIMS. **Gedaw** est un entrepôt dédié à l'annotation fonctionnelle à partir de données d'expression du transcriptome hépatique [Guérin, Marquet et al. 2005]. Il met en œuvre une intégration sémantique horizontale et verticale par des règles de correspondance. Son principal atout est la qualité des données contenues. L'intégration du schéma se fait par la détection d'analogies structurelles. Les sources intégrés sont cependant limitées aux formats structurés (relationnel : UMLS, GO, etc.) et semi-structurés (XML : GenBank, etc.). Une intégration verticale est aussi réalisée au travers de règles de correspondance donnant lieu à des regroupements. Ces règles sont éditables par un expert.

GIMS est un entrepôt orienté objet dédié à l'annotation du génome [Cornell, Paton et al. 2001]. L'intégration est horizontale et verticale. Il intègre différentes sources sur le génome, le transcriptome, et les interactions protéines-protéines, entre autres. Ses avantages sont la mise à disposition d'outils d'analyse et de requêtes graphiques avancées. L'interface utilisateur a la particularité de s'appuyer sur le modèle objet comme point d'entrée pour la navigation.

AceDB (a *C. elegans* database)est un système de gestion de base de données orienté objet mais surtout orienté vers l'intégration de données génomiques [Stein, Sternberg et al. 2001]. Il a été initialement conçu pour un projet de séquençage de l'organisme *Caenorhabditis Elegans*, un ver commun qui mesure 1 mm de long. L'objectif principal de ce projet est de fournir un système améliorant l'évolutivité et l'extensibilité du schéma. Ce système est devenu un système d'intégration complet comme en témoignent plus de cinquante bases de données qui reposent dessus. Il propose par ailleurs des outils graphiques moins souples et adaptables que le SGBD lui-même. Il est possible de l'utiliser notamment sur le site actuellement dédié à *C Elegans*, WormBase.

A.2.2 Médiateur (vues virtuelles)

Dans le contexte général de l'informatique, la communauté dispose déjà de nombreux environnements de médiation. Quelques systèmes de références sont : **Amos** [Fahl, Risch et al. 1993], **Tsimmis** [Chawathe, Garcia-Molina et al. 1994], **Disco** [Tomasic, Raschid et al. 1996], et **Garlic** [Tork Roth, Arya et al. 1996]. On peut globalement dissocier trois types d'environnement :

- les approches classiques qui s'appuient sur le modèle relationnel ou objet et sur des standards comme SQL, OQL, etc.
- les approches issues de la recherche orientées vers l'intelligence artificielles (logique de descriptions, logiques modales, règles, etc.)
- les approches orientées vers le Web et plus particulièrement sur le langage XML

Médiation Relationnelle

Discovery Link est un système commercial conçu par IBM [Haas, Schwarz et al. 2001]. Il repose sur un modèle relationnel-objet et sur le serveur de la même organisation, DB2. Il permet

d'interroger une base de données virtuelle directement en SQL. Il est notamment reconnu pour ses excellentes performances. En effet, il propose un système d'optimisation basé sur les coûts efficace et une mémoire cache qui conserve localement une partie des données. L'intégration est syntaxique : dans un premier temps, les données des sources locales sont traduites par des algorithmes dans le paradigme relationnel exploitable par le système (cette procédure s'apparente à une approche GLAV). Par la suite, on établi des correspondances dans les adaptateurs entre le schéma relationnel généré et le schéma global. Actuellement, Discovery Link évolue vers la gamme de produits du nom de Websphere Information Integrator. **Birn-M** est un médiateur relationnel dédié aux neurosciences [Gupta, Ludascher et al. 2003], produit par l'université de San Diego. Il suit une approche GAV. Il contient une connaissance du domaine sous forme de règles logiques et stocke la totalité des schémas locaux et des correspondances. **Isys** et **BioDataServer** suivent une approche GAV avec une gestion dynamique de l'intégration [Siepel, Farmer et al. 2001; Freier, Hofestädt et al. 2002]. Cette intégration repose sur une description précise des sources et visant à respecter une autonomie maximale des sources. **SEMEDA** (SEmantic MEta DAtabase) est un médiateur architecturé sous trois tiers [Kohler and Schulze-Kremer 2002; Köhler, Philippi et al. 2003]: le premier tiers s'appuie sur un SGBDR (Oracle 8i) pour stocker les métadonnées des sources. Pour se connecter aux sources, Semeda utilise de façon privilégiée JDBC. Dans le cas où les sources ne sont pas stockées dans des bases de données relationnelles (fichiers à plats, HTML, etc.), Semeda utilise BioDataServer comme adaptateur syntaxique. Il envisage aussi d'intercaler des systèmes intermédiaires comme Discovery Link. L'intégration est basée sur une ontologie (ou plusieurs), et Semeda se définit même comme un système pour intégrer des données ou concevoir et maintenir collaborativement des ontologies. Concernant l'interface utilisateur, la génération dynamique de pages Web est réalisée à l'aide de JSP.

Médiation orientée objet

TSIMMIS *(The Stanford-IBM Manager of Multiple Information Sources)* est un des premiers systèmes médiateurs orienté objet [Chawathe, Garcia-Molina et al. 1994]. Il résulte d'une collaboration entre l'université de Stanford et IBM. Il repose sur le paradigme objet, utilisant OEM (Object Exchange Model) pour spécifier le schéma et OEM-QL comme langage de requête. Il suit une approche GAV. **TINet** (Target Informatics Net) est un système qui repose sur le langage OPM, un langage voisin du standard OQL de l'ODMG avec des extensions liées aux problématiques scientifiques [Eckman, Kosky et al. 2001]. Il se focalise sur les données génomiques et suit une approche GAV. **K2** (anciennement **Kleisli**) est le système développé par l'université de Pennsylvanie [Davidson, Crabtree et al. 2001]. Il repose sur une approche GAV. Conçu en Java, il utilise le standard de modélisation objet ODL et le langage de requête OQL. Il propose de plus un langage déclaratif de haut niveau, K2ML, pour spécifier les transformations de schémas. Il propose un mécanisme de polymorphisme de types offrant plus de souplesse vis-à-vis de l'évolution des sources. Du point de vue des performances, il est doté d'un optimiseur de requêtes et permet de stocker les résultats (matérialisation). Il permet d'effectuer des requêtes de jointure entre les sources. **OPM*QS** suit une approche voisine de K2. Il est utilisé dans Genome Database (GDB) et Genome Sequence Database (GSDB). **BioZoom** suit aussi une approche objet-relationnelle. Il s'intéresse en particulier aux algorithmes d'optimisation du routage des requêtes basé sur une représentation des *« capacités des sources »*.

Médiation à bases de règles et de logiques

Les systèmes sont nombreux, principalement issus de la recherche. Sans nous étendre, voici une liste rapide des projets les plus cités. Datalog est le langage de requête le plus fréquent en matière de bases de données déductives, c'est un sous ensemble de Prolog (un langage de programmation logique)[ISO]. Il est utilisé par **Razor** [Friedman and Weld 1997] et **SoftBot** [Etzioni and Weld 1994] et étendu dans les projets **InfoMaster** [Genesereth, Keller et al. 1997] et **Information Manifold** [Kirk, Levy et al. 1995]. InfoMaster permet de spécifier des contraintes tandis qu'Information Manifold permet de représenter les atomes de formules logiques comme des classes. **Hermes** [Subrahmanian, Adali et al. 1995] utilise un schéma global (GAV) qui ne se

veut pas modéliser sémantiquement le domaine. **P/FDM** (Prolog/Functional Data Model) est l'un des rares systèmes appliqué aux données biologiques [Embury 1994]. Il se focalise sur la structure des protéines. Il est implémenté en Prolog et orienté objet. Le langage de modélisation est FDM et le langage de requête DAPLEX, un langage reconnu intuitif. D'autres systèmes utilisent les logiques de descriptions. **Sims** [Arens and Knoblock 1993] repose, par exemple, sur le langage Loom, **Observer** sur Classic [Mena, Kashyap et al. 1996] et **Momis** sur ODL-I3 [Beneventano, Bergamaschi et al. 2000], un sous-ensemble de OQL. **Picsel** (Production d'Interface à bases de Connaissances pour des Services en Ligne) s'applique au domaine du tourisme. Il exploite le langage Carin basé sur une logique de description (ALN) et réunissant les avantages des règles et des classes. Il repose sur une approche LAV. **MKBeem** descend de PICSEL et correspond à l'application de la même approche dans le contexte du commerce en ligne.

Concernant les données biologiques, les deux projets les plus connus sont Tambis et Baciis. **Tambis** (*« Transparent Access to Multiple Biological Information Source »*) possède une ontologie TaO (approche GAV) pour l'intégration [Baker, Brass et al. 1998]. TaO est formalisé dans la logique de description GRAIL. Cette ontologie est utilisée à la fois pour l'intégration des schémas mais aussi comme vocabulaire contrôlé pour l'intégration même des termes employés dans les sources locales. Tambis exploite le langage CPL de Kleisli pour la spécification des adaptateurs. **Baciis** (*« Biological and chemical information integration system »*) repose aussi sur une ontologie BAO (approche GAV). BAO est formalisée en Loom, un langage moins expressif mais moins coûteux (en calcul). On retrouve enfin quelques expériences exploitant les logiques modales et contextuelles [Farquhar, Dappert et al. 1995]. **Kind** (Knowledge-based Integration of Neuroscience Data sources) [Gupta, Ludäscher et al. 2002] emploie F-Logic [Kifer, Lausen et al. 1995] dans le contexte de données semi-structurées .

Approches décentralisées

Il existe enfin des approches virtuelles décentralisées prenant en compte l'évolution des technologies et des communautés et qui restent plus proches de l'approche fédérée des bases de données. Elles reposent sur le paradigme multi-agent (IGC-GID)[Burger, Link et al. 1997], grille [Budura, Cudré-Mauroux et al. 2007], et pair à pairs [Gribble, Halevy et al. 2001; Ives, Halevy et al. 2004; Tatarinov and Halevy 2004].

A.2.3 Approches semi-structurées (XML)

Pour garantir une conservation de la sémantique des données, le modèle doit reposer sur un paradigme aussi structurant que ceux des sources locales. On constate ainsi que les approches les plus courantes sont les approches relationnelles, objet et logiques. Qu'il s'agisse d'intégration matérialisée ou non, certaines approches reposent sur le XML ou éventuellement d'autres approches semi-structurées [Achard, Vaysseix et al. 2001; Baril 2003]. Les approches semi structurées se situent généralement dans deux contextes.

Le premier est le contexte général du Web Sémantique. Les documents sont généralement du texte non structurés. Le XML est donc adapté à ce type de contenu. Un projet d'entrepôt célèbre est par exemple **Xylème** [Delobel, Reynaud et al. 2003]. Concernant les approches médiateur, des projets de référence sont **Mix** (*Mediation of Information using Xml* [Baru, Gupta et al. 1999]) **C-Web** et **Piazza** qui exploite XML et ses dérivés RDF et OWL dans une approche décentralisée de *« pair à pair » [Ives, Halevy et al. 2004]*.

Dans le contexte de la bioinformatique, le XML est fréquemment utilisé comme format de d'échange de données. Il n'est cependant pas suffisamment expressif ou pratique pour exprimer des modèles équivalents à des schémas objets ou relationnels. Les expériences sont assez peu nombreuses en matière de système d'intégration, les sources étant généralement structurées à l'aide paradigme plus expressifs. Cette approche XML est plus fréquente dans les systèmes à base de liens (cf. section suivante) comme BioMediator. **gRNA** se définit comme un outil pour concevoir et déployer des entrepôts distribués [Bhouwmick, Cruz et al. 2002]. Il est centré sur

des données génomiques. **BioBroker** adopte le XML dans une approche médiateur (LAV). Rappelons aussi le système KIND présenté précédemment parmi les systèmes basés sur une logique modale.

Annexe B UML

B.1 Diagrammes de classes

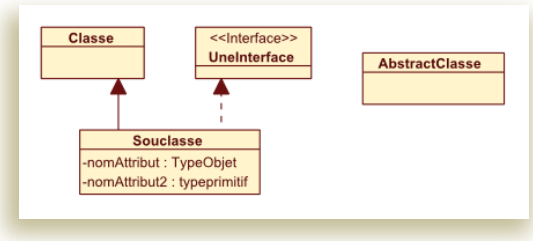


Figure B.2 – Classe, interface, attribut et relation de généralisation.

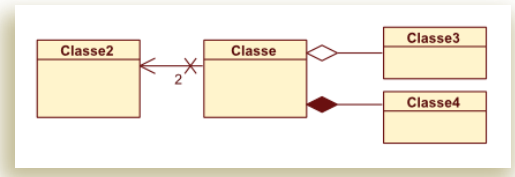


Figure B.3 – Association, aggrégation et composition.

Le diagramme de classe (*« static diagram »*) d'UML est utilisé ici uniquement pour la modélisation des types abstraits et non des instances. Les rectangles sont donc des types abstraits (figure b.2). Les classes sont représentées de façon simple, le stéréotype « interface » indique qu'il s'agit d'une interface. Les classes abstraites sont préfixées par « Abstract ». Les principaux attributs et leurs types sont représentés dans les rectangles des classes. Les méthodes ne sont pas représentées. Les flèches à pointe pleines représentent l'héritage de propriétés. Lorsqu'elles sont continues il s'agit d'héritage ; lorsqu'elles sont discontinues, il s'agit d'implémentation d'une interface.

Dans la figure b.3, trois relations supplémentaires entre classes ou interfaces sont représentées. La flèche (et la croix) représente l'association entre deux classes. Dans cette figure, cela signifie qu'une instance de `Classe` contient un attribut référençant une instance de `Classe2`. Une valeur peut être ajoutée à la relation : ici, le 2 indique qu'en fait `Classe` contient deux attributs référençant chacun une instance, identique ou non, de `Classe2`.

Le losange indique une relation de d'agrégation. Une instance de `Classe3` et `Classe4` va contenir une instance de `Classe`. Les cardinalités sont alors indiqués par un chiffre, et « `*` » ou « `n` », pour représenter une valeur multiple indéterminée. Lorsque le losange est plein, on parle de composition (ou agrégation forte) : la durée de vie de l'instance contenue est dépendante de celle du contenant. Autrement dit, la destruction en mémoire d'une instance de `Classe4` entraîne la suppression en mémoire de toutes les instances de `Classe` qui la composent.

B.2 Diagrammes d'activités

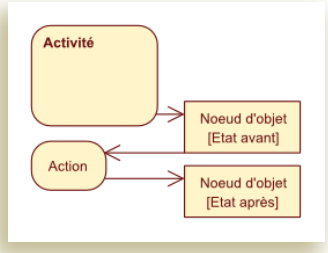


Figure B.4 – Trois éléments : activité, action et objet, liés par des flux de contrôle et flux d'objets. L'objet peut posséder un état qui évolue.

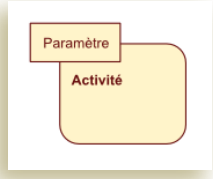


Figure B.5 – Activité paramétrée.

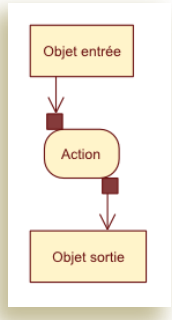


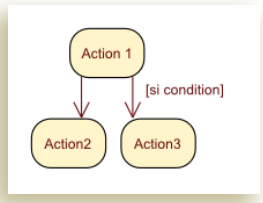
Figure B.6 – « Pin », le paramètre de l'action.

Le diagramme d'activité repose sur trois éléments de base (figure b.4): l'activité, l'action et l'objet. L'activité et l'action peuvent s'interpréter comme des procédures, mais de deux façons distinctes. Certains considèrent l'action comme atomique, et l'activité comme non atomique (une composition d'actions et/ou d'activités). D'autres définissent l'action comme une procédure ; L'activité est alors permanente et évènementielle, voisine de la notion de service ou d'écouteur. Nous avons préféré cette seconde interprétation.

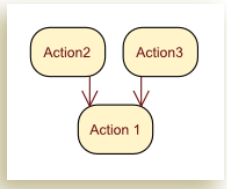
L'objet décrit un élément ou un ensemble de données et peut être caractérisé par un état (entre crochets). Ces schémas n'ont pas pour objectif de correspondre exactement à l'implémentation réalisée dans un langage à objets. L'objet du diagramme n'a donc pas la même sémantique que celle d'un langage à objet, et ne correspond pas forcément à l'instance d'une classe déterminée. En outre, dans notre cas, la taille des données implique généralement le recours à des techniques de persistance et impliquent l'instanciation de multiples objets, de structures de données et de collections complexes. De plus, ces diagrammes visant à décrire l'ordonnancement de certaines opérations et procédures, nous faisons abstraction des différents niveaux de mémoire (centrale, secondaire) et leur localisation ou structuration (disque dur, fichier ou SGBDR, etc.). Notons qu'au niveau de granularité de ces diagrammes, les objets sont généralement volumineux au point d'être sauvegardés dans une mémoire dite persistante.

Les flèches structurent l'ordonnancement des procédures et spécifient leurs dépendances directes et indirectes. On distingue les flux de contrôle qui relient des actions (et activités) entre-elles, et les flux d'objets qui relient les actions (et activités) à des objets produits, modifiés ou utilisés. L'orientation de ces flux permet de déduire assez simplement la chronologie. Par abus de syntaxe et pour alléger les schémas, nous n'avons volontairement pas représenté les états initiaux et finaux.

Les actions et les activités peuvent être paramétrées. La figure b.5 montre le schématisme pour l'activité. Concernant les objets (figure b.6), le symbole de paramètre d'entrée ou de sortie est un petit carré (« *pin* »). La différence entre un paramètre d'entrée et un objet directement relié à l'action est principalement justifiée par la durée de vie de l'objet. Celle du « *pin* » est normalement restreinte à l'exécution du diagramme, l'objet relié directement à l'action peut être persistant en dehors du diagramme.

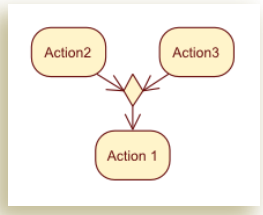


(a) En sortie

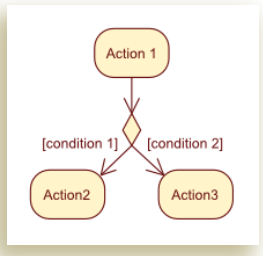


(b) En entrée

Figure B.7 – Alternatives de flux de contrôle.



(a) Fusion (merge)

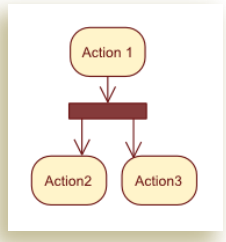


(b) Décision

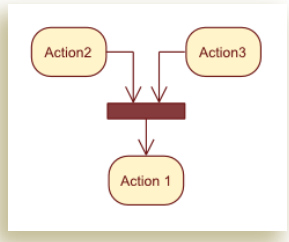
Figure B.8 – Nœud de fusion et décision pour des spécifications formelles complexes.

Il est possible de formaliser les flux de contrôle et d'objets avec plus de précision. Si rien n'est précisé, plusieurs flux qui partent d'une action, ou qui arrivent sur une même action signifient une alternative (figure b.7). Dans l'exemple (figure b.7-a) ci-dessus, l'action 1 peut soit appeler l'exécution de l'action 2, soit celle de l'action 3, mais pas les deux simultanément. Il s'agit donc d'un *« ou exclusif »*. Plusieurs exécutions de l'action 1 n'aboutissent pas obligatoirement à la même action. Il est possible de spécifier des *gardes* (entre crochets), des conditions sur les flux de contrôle. Ces conditions sont nécessaires. Par exemple, pour qu'A1 appelle A3, il faut nécessairement que la condition soit remplie. Cependant, la validité de cette condition n'interdit pas l'appel d'A2 par A1.La figure b.7-b possède la même signification. Deux actions qui débouchent sur une troisième n'impliquent pas une synchronisation. L'action 1 s'exécute à la demande de l'action 2 ou de l'action 3. Il n'est pas nécessaire d'attendre que ces deux actions antérieures soient toutes deux réalisées. La sémantique est analogue concernant les objets.

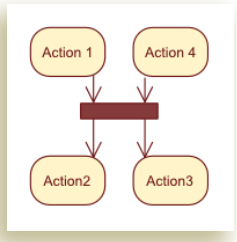
Le formalisme des nœuds de fusion (figure b.8-a) et de décision (figure b.8-b) possède une signification assez proche des contrôles de flux multiples avec des gardes. Ils notifient plus intuitivement l'étape de décision ou de fusion, dans respect des pratiques d'un large ensemble de formalisme. Leur apport essentiel est alors la possibilité de multiplier et combiner les décisions entre deux actions ou activités.



(a) Fork (fourchette)



(b) Join (jointure)



(c) Combinaison des deux.

Figure B.9 – Synchronisation des flux.

Les deux représentations précédentes ne permettent que des cheminements alternatifs. La fourchette (figure b.9-a) décrit une action qui produit plusieurs objets, appelle plusieurs actions ou activités. La jointure (figure b.9-b) permet réciproquement de représenter la synchronisation des entrées d'une procédure ou d'un objet : L'action 1, pour s'exécuter, attend que tous les objets soient arrivés (dans le bon état éventuellement) ou que les actions antérieures se soient toutes produites. La figure b.9-c montre comment il est possible de combiner ces deux synchronisations simultanément.

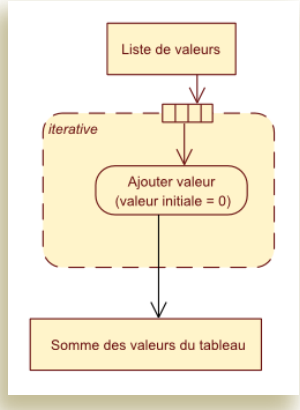


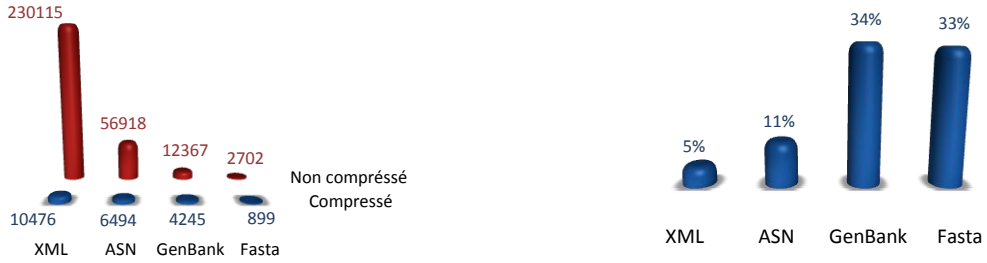
Figure B.10 – Région d'expansion. Cet exemple représente le calcul de la somme des valeurs d'une collection.

Enfin, le dernier aspect des diagrammes d'activité qui concerne ce mémoire est la région d'expansion (figure b.10). Une région d'expansion est délimitée par une ligne discontinue. Elle est paramétrée par une collection d'objets ( ). Dès lors, le contenu de la région d'expansion est exécuté pour chaque élément de la collection. Une région d'expansion possède un attribut (en haut à gauche de la région). Il peut être de trois types, « Iterative » lorsque le l'exécution est itérative (séquentielle), « Parallel » lorsqu'elle est parallélisable, et « Streaming » lorsqu'il s'agit de flux continu de données. L'exemple ci-dessus représente le calcul de la somme des valeurs d'un tableau.

Annexe C Exemples

C.1 Formats de séquences nucléotidiques

Nous avons inclus ici quelques formats les plus courants disponibles notamment dans GenBank (Entrez Nucleotide). Après une recherche sur le mot clé « G-CSF », nous avons sélectionné le gène *« Homo sapiens colony stimulating factor 1 (macrophage) (CSF1) »* identifié par les numéros d'accession suivants : `NM_172212.1` et `GI:27262666`. Les sections suivantes montre la structure du contenu. Certains de ces contenus sont plus expressifs et structurés, mais aussi plus volumineux et moins lisibles par l'utilisateur sans mise en forme.



Volume du fichier *Taux de compression*

Figure C.11 – A gauche, le graphique montre pour quatre formats de fichier courant la taille du fichier pour la séquence NM_172212.1. Cette taille est donnée avec et sans compression. A droite le graphique montre le taux de compression pour ces mêmes formats et pour le même fichier exemple. On constate que les formats les plus structurés sont jusqu'à 100 fois plus volumineux que ceux les moins descriptifs. La balisage employer pour la structuration est cependant facilement compressible. Par conséquent, ces formats qui ont les meilleurs taux de compression s'avère nettement moins couteux en stockage en comparaison des autres formats.

C.1.1 ASN.1

```
Seq-entry ::= set {
  level 1 ,
  class nuc-prot ,
  descr {
    source {
      genome genomic ,
      org {
        taxname "Homo sapiens" ,
        common "human" ,
        db {
          {
            db "taxon" ,
            tag
              id 9606 } } ,
        orgname {
          name
            binomial {
              genus "Homo" ,
              species "sapiens" } ,
          lineage "Eukaryota; Metazoa; Chordata; Craniata; Vertebrata;
 […]
 publications that are available for this gene. Please see the Entrez Gene
 record to access additional publications." ,
```

```
    pub {
      pub {
        pmid 17243911 ,
        article {
          title {
            name "Serum levels of granulocyte colony-stimulating factor
(G-CSF) and macrophage colony-stimulating factor (M-CSF) in pancreatic cancer
patients." } ,
          authors {
            names
              std {
                {
                  name
                    name {
                      last "Groblewska" ,
                      initials "M." } } ,
                {
                  name
                    name {
                      last "Mroczko" ,
                      initials "B." } } ,
                {
                  name
                    name {
                      last "Wereszczynska-Siemiatkowska" ,
                      initials "U." } } ,
                {
                  name
                    name {
                      last "Mysliwiec" ,
                      initials "P." } } ,
                {
                  name
                    name {
                      last "Kedra" ,
                      initials "B." } } ,
                {
                  name
                    name {
                      last "Szmitkowski" ,
                      initials "M." } } } ,
            affil
              str "Department of Biochemical Diagnostics, Medical University,
Bialystok, Poland." } ,
          from
            journal {
              title {
                iso-jta "Clin. Chem. Lab. Med." ,
                ml-jta "Clin Chem Lab Med" ,
                issn "1434-6621" } ,
              imp {
                date
                  std {
                    year 2007 } ,
                volume "45" ,
                issue "1" ,
                pages "30-34" ,
                language "eng" ,
                pubstatus ppublish ,
                history {
                  {
                    pubstatus pubmed ,
                    date
                      std {
                        year 2007 ,
                        month 1 ,
                        day 25 ,
                        hour 9 ,
                        minute 0 } } ,
                  {
                    pubstatus medline ,
                    date
                      std {
                        year 2007 ,
                        month 3 ,
                        day 21 ,
                        hour 9 ,
                        minute 0 } } } } } ,
```

```
          ids {
            doi "10.1515/CCLM.2007.025" ,
            pubmed 17243911 } } } ,
      comment "GeneRIF: significantly higher levels of macrophage-colony
 stimulating factor is associated with pancreatic cancer" } ,
[…]
    update-date
      std {
        year 2007 ,
        month 3 ,
        day 25 } } ,
  seq-set {
    seq {
      id {
        other {
          accession "NM_172212" ,
          version 1 } ,
        gi 27262666 } ,
      descr {
        molinfo {
          biomol mRNA } ,
        title "Homo sapiens colony stimulating factor 1 (macrophage) (CSF1),
 transcript variant 4, mRNA" ,
        create-date
          std {
            year 2002 ,
            month 12 ,
            day 19 } } ,
      inst {
        repr raw ,
        mol rna ,
        length 2545 ,
        seq-data
          ncbi2na
'8A9E94B8A7695AA02E02FE5EAD7769948967759354A126B9A57696A9951D6492526262626262A6961995A5A854
9E56CE1665A9965A99E5754613A7A9D5E7BEFADEDD7A624A2CD1628AEDA2C7B25138FA8BA11792DDE49A78F84B4
8E885DB940F13F8BFB214A04BE02352EE717C20A4FDD7AC42130E8A114E67D22304554394D94FB9279281DDFE29
E089E7D1428F38224E10A5E6D607F738845DD4BE7A20AD083B7F0E01020DD7E10A1E833FD24207904127F9E0E75
2508EEB850978F90797B154094D5C92E15A5DEDD574D255D955D4E957BA7A7E17A8A1DE2A078A927577E5EB8925
5E444BA35292E509269454A245E5227F896522154BED0A124536BA745125D9577B6A97D055A8E8A33DF87790EA4
70FAB54820977A2294B88F56C550A8489FD55D4A528AA924E48489594854907D774937DD4757934902A50496923
B07AC497E54ABA956E2951E94A1E83445552084853537795E7488555A2529DD528DD34799552A5D24157515DDE74
9449FD48251D76A49B9E55FAA27A2A4A28914A8DA289564894828A1490B82A9252957956FF075BD7F87844A5388
A4B58A8D7525649D4A2DEDF5179EB952ED35EB7E7A5B68A5DFBDC4AE8A69A2534225D22268F757E890522A4955E
1D28E1212BA0794BB22A0F70855D14D7A111DBFB43B577803B8654956844B1D48EFB785274888B12EA1EF17D7E3
3A12CF7DCFEE48F08F93CBFFF7C107934C7BED33BE25EEBDCC01572F53F54C07DED0948537715EC7E841F07FFC1
40B92FCEF45FEF02517FBAFDE54D178171E0BEEE035C3DED375B25754BEE5D7913E38B979EFB7F953BEF8EC9EE1
5CFBD7455E555941549E95177D55D515095125254D2825F5E9F75105F787B7FD2D3955E77FECFE9C32CCD0FE47C
0'H } ,
      annot {
        {
          data
            ftable {
              {
                data
                  gene {
                    locus "CSF1" ,
                    desc "colony stimulating factor 1 (macrophage)" ,
                    syn {
                      "MCSF" ,
                      "MGC31930" } } ,
                location
                  int {
                    from 0 ,
                    to 2544 ,
                    strand plus ,
                    id
                      gi 27262666 } ,
                dbxref {
                  {
                    db "GeneID" ,
                    tag
                      id 1435 } ,
                  {
[…]
                db "CCDS" ,
                tag
                  str "CCDS816.1" } } } } } } }
```

C.1.2 Fasta

```
>gi|27262666|ref|NM_172212.1| Homo sapiens colony stimulating factor 1 (macrophage) (CSF1),
transcript variant 4, mRNA
GAGGGCTGGCCAGTGAGGCTCGGCCCGGGGAAAGTGAAAGTTTGCCTGGGTCCTCTCGGCGCCAGAGCCGCTCTCCGCATCCCAGGACAGC
GGTGCGGCCCTCGGCCGGGGCGCCCACTCCGCAGCAGCCAGCGAGCGAGCGAGCGAGCGAGGGCGGCCGACGCGCCCGGCCGGGACCCAGC
TGCCCGTATGACCGCGCCGGGCGCCGCCGGGCGCTGCCCTCCCACGACATGGCTGGGCTCCCTGCTGTTGTTGGTCTGTCTCCTGGCGAGC
AGGAGTATCACCGAGGAGGTGTCGGAGTACTGTAGCCACATGATTGGGAGTGGACACCTGCAGTCTCTGCAGCGGCTGATTGACAGTCAGA
TGGAGACCTCGTGCCAAATTACATTTGAGTTTGTAGACCAGGAACAGTTGAAAGATCCAGTGTGCTACCTTAAGAAGGCATTTCTCCTGGT
ACAAGACATAATGGAGGACACCATGCGCTTCAGAGATAACACCCCCAATGCCATCGCCATTGTGCAGCTGCAGGAACTCTCTTTGAGGCTG
AAGAGC
TGCTTCACCAAGGATTATGAAGAGCATGACAAGGCCTGCGTCCGAACTTTCTATGAGACACCTCTCCAGTTGCTGGAGACATTGATGAGTG
C
[…]
CTGCTGTTGTCTTTGCCCATGTTGTTGATGTAGCTGTGACCCTATTGTTCCTCACCCCTGCCCCCCGCCAACCCCAGCTGGCCCACCTCTT
C
CCCCTCCCACCCAAGCCCACAGCCAGCCCATCAGGAAGCCTTCCTGGCTTCTCCACAACCTTCTGACTGTCTTTTCAGTCATGCCCCCTGC
TCTTTTGTATTTGGCTAATAGTATATCAATTTGCACTT
```

C.1.3 XML

```
<?xml version="1.0"?>
<!DOCTYPE Seq-entry PUBLIC "-//NCBI//NCBI Seqset/EN"
"http://www.ncbi.nlm.nih.gov/dtd/NCBI_Seqset.dtd">
<Seq-entry>
  <Seq-entry_set>
    <Bioseq-set>
      <Bioseq-set_level>1</Bioseq-set_level>
      <Bioseq-set_class value="nuc-prot"/>
      <Bioseq-set_descr>
        <Seq-descr>
          <Seqdesc>
            <Seqdesc_source>
              <BioSource>
                <BioSource_genome value="genomic">1</BioSource_genome>
                <BioSource_org>
                  <Org-ref>
                    <Org-ref_taxname>Homo sapiens</Org-ref_taxname>
                    <Org-ref_common>human</Org-ref_common>
                    <Org-ref_db>
                      <Dbtag>
                        <Dbtag_db>taxon</Dbtag_db>
                        <Dbtag_tag>
                          <Object-id>
                            <Object-id_id>9606</Object-id_id>
                          </Object-id>
                        </Dbtag_tag>
                      </Dbtag>
                    </Org-ref_db>
                    <Org-ref_orgname>
                      <OrgName>
                        <OrgName_name>
                          <OrgName_name_binomial>
                            <BinomialOrgName>
                              <BinomialOrgName_genus>Homo</BinomialOrgName_genus>
                              <BinomialOrgName_species>sapiens</BinomialOrgName_species>
                            </BinomialOrgName>
                          </OrgName_name_binomial>
                        </OrgName_name>
                        <OrgName_lineage>Eukaryota; Metazoa; Chordata; Craniata;
Vertebrata; Euteleostomi; Mammalia; Eutheria; Euarchontoglires; Primates; Haplorrhini;
Catarrhini; Hominidae; Homo</OrgName_lineage>
                        <OrgName_gcode>1</OrgName_gcode>
                        <OrgName_mgcode>2</OrgName_mgcode>
                        <OrgName_div>PRI</OrgName_div>
                      </OrgName>
                    </Org-ref_orgname>
                  </Org-ref>
                </BioSource_org>
                <BioSource_subtype>
                  <SubSource>
                    <SubSource_subtype value="chromosome">1</SubSource_subtype>
                    <SubSource_name>1</SubSource_name>
```

```
                  </SubSource>
                  <SubSource>
                    <SubSource_subtype value="map">2</SubSource_subtype>
                    <SubSource_name>1p21-p13</SubSource_name>
                  </SubSource>
                </BioSource_subtype>
              </BioSource>
            </Seqdesc_source>
          </Seqdesc>
          <Seqdesc>
            <Seqdesc_comment>~Summary: The protein encoded by this gene is a cytokine that
controls the production, differentiation, and function of macrophages. The active form of
the protein is found extracellularly as a disulfide-linked homodimer, and is thought to be
produced by proteolytic cleavage of membrane-bound precursors. The encoded protein may be
involved in development of the placenta. Four transcript variants encoding three different
isoforms have been found for this gene.</Seqdesc_comment>
          </Seqdesc>
          <Seqdesc>
            <Seqdesc_comment>~Transcript Variant: This variant (4) differs in the 3' UTR
compared to variant 1. Both variants 1 and 4 encode isoform a.</Seqdesc_comment>
          </Seqdesc>
          <Seqdesc>
            <Seqdesc_user>
              <User-object>
                <User-object_type>
                  <Object-id>
                    <Object-id_str>RefGeneTracking</Object-id_str>
                  </Object-id>
                </User-object_type>
                <User-object_data>
                  <User-field>
                    <User-field_label>
                      <Object-id>
                        <Object-id_str>Status</Object-id_str>
                      </Object-id>
                    </User-field_label>
                    <User-field_data>
                      <User-field_data_str>Reviewed</User-field_data_str>
                    </User-field_data>
                  </User-field>
                  <User-field>
                    <User-field_label>
                      <Object-id>
                        <Object-id_str>Assembly</Object-id_str>
                      </Object-id>
                    </User-field_label>
                    <User-field_data>
                      <User-field_data_fields>
                        <User-field>
                          <User-field_label>
                            <Object-id>
                              <Object-id_id>0</Object-id_id>
                            </Object-id>
                          </User-field_label>
                          <User-field_data>
                            <User-field_data_fields>
                              <User-field>
                                <User-field_label>
                                  <Object-id>
                                    <Object-id_str>accession</Object-id_str>
                                  </Object-id>
                                </User-field_label>
                                <User-field_data>
                                  <User-field_data_str>M27087.1</User-field_data_str>
                                </User-field_data>
                              </User-field>
                              <User-field>
                                <User-field_label>
                                  <Object-id>
                                    <Object-id_str>gi</Object-id_str>
                                  </Object-id>
[…]
            <Seqdesc_comment>~Publication Note:  This RefSeq record includes a subset of
the publications that are available for this gene. Please see the Entrez Gene record to
access additional publications.</Seqdesc_comment>
          </Seqdesc>
          <Seqdesc>
            <Seqdesc_pub>
```

```
              <Pubdesc>
                <Pubdesc_pub>
                  <Pub-equiv>
                    <Pub>
                      <Pub_pmid>
                        <PubMedId>17243911</PubMedId>
                      </Pub_pmid>
                    </Pub>
                    <Pub>
                      <Pub_article>
                        <Cit-art>
                          <Cit-art_title>
                            <Title>
                              <Title_E>
                                <Title_E_name>Serum levels of granulocyte colony-
stimulating factor (G-CSF) and macrophage colony-stimulating factor (M-CSF) in pancreatic
cancer patients.</Title_E_name>
                              </Title_E>
                            </Title>
                          </Cit-art_title>
                          <Cit-art_authors>
                            <Auth-list>
                              <Auth-list_names>
                                <Auth-list_names_std>
                                  <Author>
                                    <Author_name>
                                      <Person-id>
                                        <Person-id_name>
                                          <Name-std>
                                            <Name-std_last>Groblewska</Name-std_last>
                                            <Name-std_initials>M.</Name-std_initials>
                                          </Name-std>
[…]
                                      <Title_E_iso-jta>Clin. Chem. Lab. Med.</Title_E_iso-
jta>
                                    </Title_E>
                                    <Title_E>
                                      <Title_E_ml-jta>Clin Chem Lab Med</Title_E_ml-jta>
                                    </Title_E>
                                    <Title_E>
                                      <Title_E_issn>1434-6621</Title_E_issn>
                                    </Title_E>
                                  </Title>
                                </Cit-jour_title>
                                <Cit-jour_imp>
                                  <Imprint>
                                    <Imprint_date>
                                      <Date>
                                        <Date_std>
                                          <Date-std>
                                            <Date-std_year>2007</Date-std_year>
                                          </Date-std>
                                        </Date_std>
                                      </Date>
                                    </Imprint_date>
                                    <Imprint_volume>45</Imprint_volume>
                                    <Imprint_issue>1</Imprint_issue>
                                    <Imprint_pages>30-34</Imprint_pages>
                                    <Imprint_language>eng</Imprint_language>
                                    <Imprint_pubstatus>
                                      <PubStatus value="ppublish">4</PubStatus>
[…]
        <Seq-entry>
          <Seq-entry_seq>
            <Bioseq>
              <Bioseq_id>
                <Seq-id>
                  <Seq-id_other>
                    <Textseq-id>
                      <Textseq-id_accession>NM_172212</Textseq-id_accession>
                      <Textseq-id_version>1</Textseq-id_version>
                    </Textseq-id>
                  </Seq-id_other>
                </Seq-id>
                <Seq-id>
                  <Seq-id_gi>27262666</Seq-id_gi>
                </Seq-id>
              </Bioseq_id>
```

```
              <Bioseq_descr>
                <Seq-descr>
                  <Seqdesc>
                    <Seqdesc_molinfo>
                      <MolInfo>
                        <MolInfo_biomol value="mRNA">3</MolInfo_biomol>
                      </MolInfo>
                    </Seqdesc_molinfo>
                  </Seqdesc>
                  <Seqdesc>
                    <Seqdesc_title>Homo sapiens colony stimulating factor 1 (macrophage)
(CSF1), transcript variant 4, mRNA</Seqdesc_title>
                  </Seqdesc>
                  <Seqdesc>
                    <Seqdesc_create-date>
                      <Date>
                        <Date_std>
                          <Date-std>
                            <Date-std_year>2002</Date-std_year>
                            <Date-std_month>12</Date-std_month>
                            <Date-std_day>19</Date-std_day>
                          </Date-std>
                        </Date_std>
                      </Date>
                    </Seqdesc_create-date>
                  </Seqdesc>
                </Seq-descr>
              </Bioseq_descr>
              <Bioseq_inst>
                <Seq-inst>
                  <Seq-inst_repr value="raw"/>
                  <Seq-inst_mol value="rna"/>
                  <Seq-inst_length>2545</Seq-inst_length>
                  <Seq-inst_seq-data>
                    <Seq-data>
                      <Seq-data_iupacna>

<IUPACna>GAGGGCTGGCCAGTGAGGCTCGGCCCGGGGAAAGTGAAAGTTTGCCTGGGTCCTCTCGGCGCCAGAGCCGCTCTCCGCATCC
CAGGACAGCGGTGCGGCCCTCGGCCGGGGCGCCCACTCCGCAGCAGCCAGCGAGCGAGCGAGCGAGCGAGGGCGGCCGACGCGCCCGGCCG
GGACCCAGCTGCCCGTATGACCGCGCCGGGCGCCGCCGGGCGCTGCCCTCCCACGACATGGCTGGGCTCCCTGCTGTTGTTGGTCTGTCTC
CTGGCGAGCAGGAGTATCACCGAGGAGGTGTCGGAGTACTGTAGCCACATGATTGGGAGTGGACACCTGCAGTCTCTGCAGCGGCTGATTG
ACAGTCAGATGGAGACCTCGTGCCAAATTACATTTGAGTTTGTAGACCAGGAACAGTTGAAAGATCCAGTGTGCTACCTTAAGAAGGCATT
TCTCCTGGTACAAGACATAATGGAGGACACCATGCGCTTCAGAGATAACACCCCCAATGCCATCGCCATTGTGCAGCTGCAGGAACTCTCT
TTGAGGCTGAAGAGCTGCTTCACCAAGGATTATGAAGAGCATGACAAGGCCTGCGTCCGAACTTTCTATGAGACACCTCTCCAGTTGCTGG
AGAAGGTCAAGAATGTCTTTAATGAAACAAAGAATCTCCTTGACAAGGACTGGAATATTTTCAGCAAGAACTGCAACAACAGCTTTGCTGA
ATGCTCCAGCCAAGATGTGGTGACCAAGCCTGATTGCAACTGCCTGTACCCCAAAGCCATCCCTAGCAGTGACCCGGCCTCTGTCTCCCCT
CATCAGCCCCTCGCCCCCTCCATGGCCCCTGTGGCTGGCTTGACCTGGGAGGACTCTGAGGGAACTGAGGGCAGCTCCCTCTTGCCTGGTG
AGCAGCCCCTGCACACAGTGGATCCAGGCAGTGCCAAGCAGCGGCCACCCAGGAGCACCTGCCAGAGCTTTGAGCCGCCAGAGACCCCAGT
TGTCAAGGACAGCACCATCGGTGGCTCACCACAGCCTCGCCCCTCTGTCGGGGCCTTCAACCCCGGGATGGAGGATATTCTTGACTCTGCA
ATGGGCACTAATTGGGTCCCAGAAGAAGCCTCTGGAGAGGCCAGTGAGATTCCCGTACCCCAAGGGACAGAGCTTTCCCCCTCCAGGCCAG
GAGGGGGCAGCATGCAGACAGAGCCCGCCAGACCCAGCAACTTCCTCTCAGCATCTTCTCCACTCCCTGCATCAGCAAAGGGCCAACAGCC
GGCAGATGTAACTGGTACAGCCTTGCCCAGGGTGGGCCCCGTGAGGCCCACTGGCCAGGACTGGAATCACACCCCCCAGAAGACAGACCAT
CCATCTGCCCTGCTCAGAGACCCCCCGGAGCCAGGCTCTCCCAGGATCTCATCACTGCGCCCCCAGGGCCTCAGCAACCCCTCCACCCTCT
CTGCTCAGCCACAGCTTTCCAGAAGCCACTCCTCGGGCAGCGTGCTGCCCCTTGGGGAGCTGGAGGGCAGGAGGAGCACCAGGGATCGGAG
GAGCCCCGCAGAGCCAGAAGGAGGACCAGCAAGTGAAGGGGCAGCCAGGCCCCTGCCCCGTTTTAACTCCGTTCCTTTGACTGACACAGGC
CATGAGAGGCAGTCCGAGGGATCCTCCAGCCCGCAGCTCCAGGAGTCTGTCTTCCACCTGCTGGTGCCCAGTGTCATCCTGGTCTTGCTGG
CCGTCGGAGGCCTCTTGTTCTACAGGTGGAGGCGGCGGAGCCATCAAGAGCCTCAGAGAGCGGATTCTCCCTTGGAGCAACCAGAGGGCAG
CCCCCTGACTCAGGATGACAGACAGGTGGAACTGCCAGTGTAGAGGGAATTCTAAGACCCCTCACCATCCTGGACACACTCGTTTGTCAAT
GTCCCTCTGAAAATGTGACGCCCAGCCCCGGACACAGTACTCCAGATGTTGTCTGACCAGCTCAGAGAGAGTACAGTGGGACTGTTACCTT
CCTTGATATGGACAGTATTCTTCTATTTGTGCAGATTAAGATTGCATTAGTTTTTTTCTTAACAACTGCATCATACTGTTGTCATATGTTG
AGCCTGTGGTTCTATAAAACCCCTAGTTCCATTTCCCATAAACTTCTGTCAAGCCAGACCATCTCTACCCTGTACTTGGACAACTTAACTT
TTTTAACCAAAGTGCAGTTTATGTTCACCTTTGTTAAAGCCACCTTTGTGGTTTCTGCCCATCACCTGAACCTACTGAAGTTGTGTGAAAT
CCTAATTCTGTCATCTCCGTAGCCCTCCCAGTTGTGCCTCCTGCACATTGATGAGTGCCTGCTGTTGTCTTTGCCCATGTTGTTGATGTAG
CTGTGACCCTATTGTTCCTCACCCCTGCCCCCCGCCAACCCCAGCTGGCCCACCTCTTCCCCCTCCCACCCAAGCCCACAGCCAGCCCATC
AGGAAGCCTTCCTGGCTTCTCCACAACCTTCTGACTGTCTTTTCAGTCATGCCCCCTGCTCTTTTGTATTTGGCTAATAGTATATCAATTT
GCACTT</IUPACna>
                      </Seq-data_iupacna>
                    </Seq-data>
                  </Seq-inst_seq-data>
                </Seq-inst>
              </Bioseq_inst>
              <Bioseq_annot>
                <Seq-annot>
                  <Seq-annot_data>
                    <Seq-annot_data_ftable>
                      <Seq-feat>
                        <Seq-feat_data>
                          <SeqFeatData>
```

```
                            <SeqFeatData_gene>
                              <Gene-ref>
                                <Gene-ref_locus>CSF1</Gene-ref_locus>
                                <Gene-ref_desc>colony stimulating factor 1
(macrophage)</Gene-ref_desc>
                                <Gene-ref_syn>
                                  <Gene-ref_syn_E>MCSF</Gene-ref_syn_E>
                                  <Gene-ref_syn_E>MGC31930</Gene-ref_syn_E>
                                </Gene-ref_syn>
[…]
                        <Object-id_str>CCDS816.1</Object-id_str>
                      </Object-id>
                    </Dbtag_tag>
                  </Dbtag>
                </Seq-feat_dbxref>
              </Seq-feat>
            </Seq-annot_data_ftable>
          </Seq-annot_data>
        </Seq-annot>
      </Bioseq-set_annot>
    </Bioseq-set>
  </Seq-entry_set>
</Seq-entry>
```

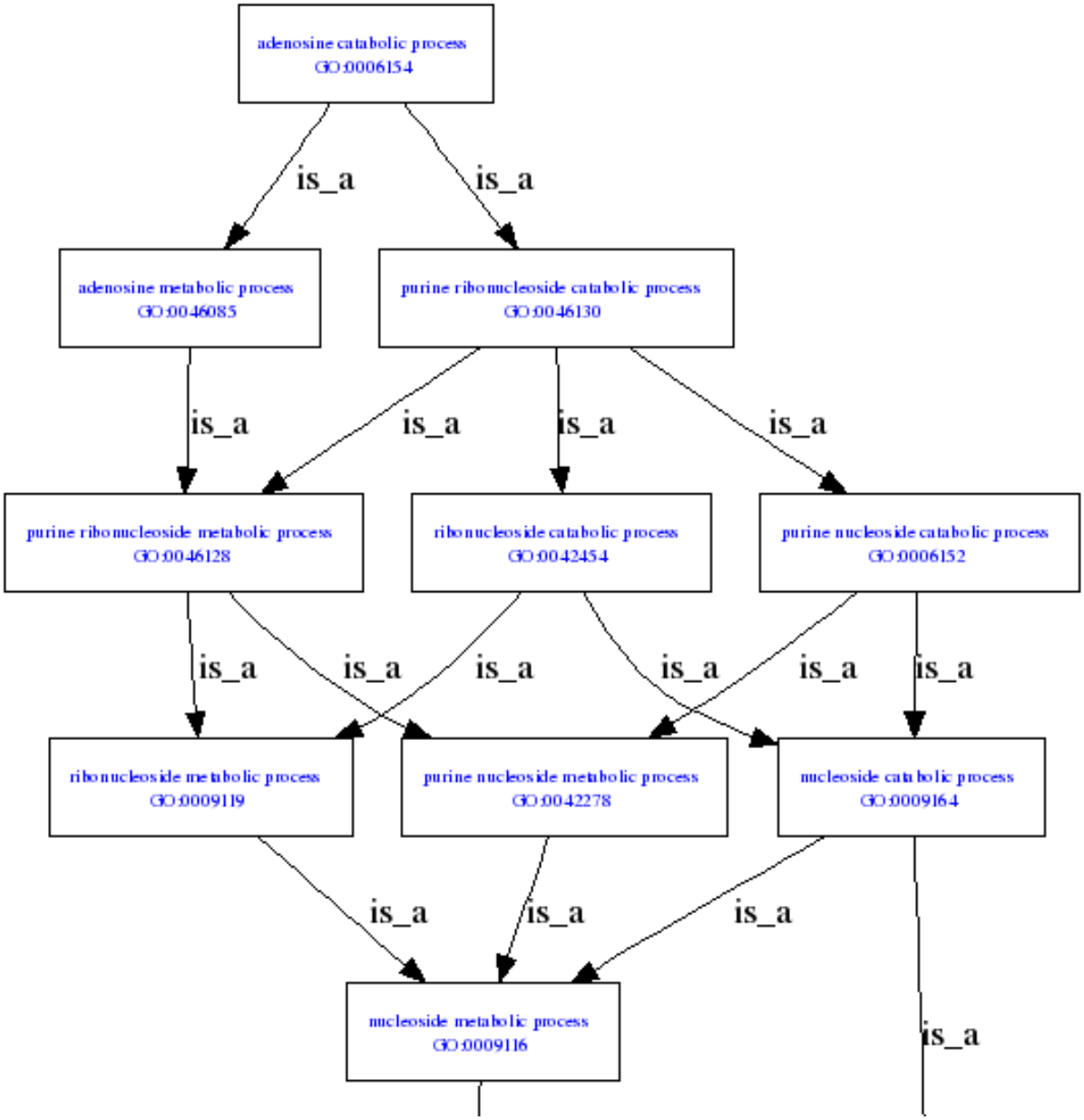
C.1.4 GenBank

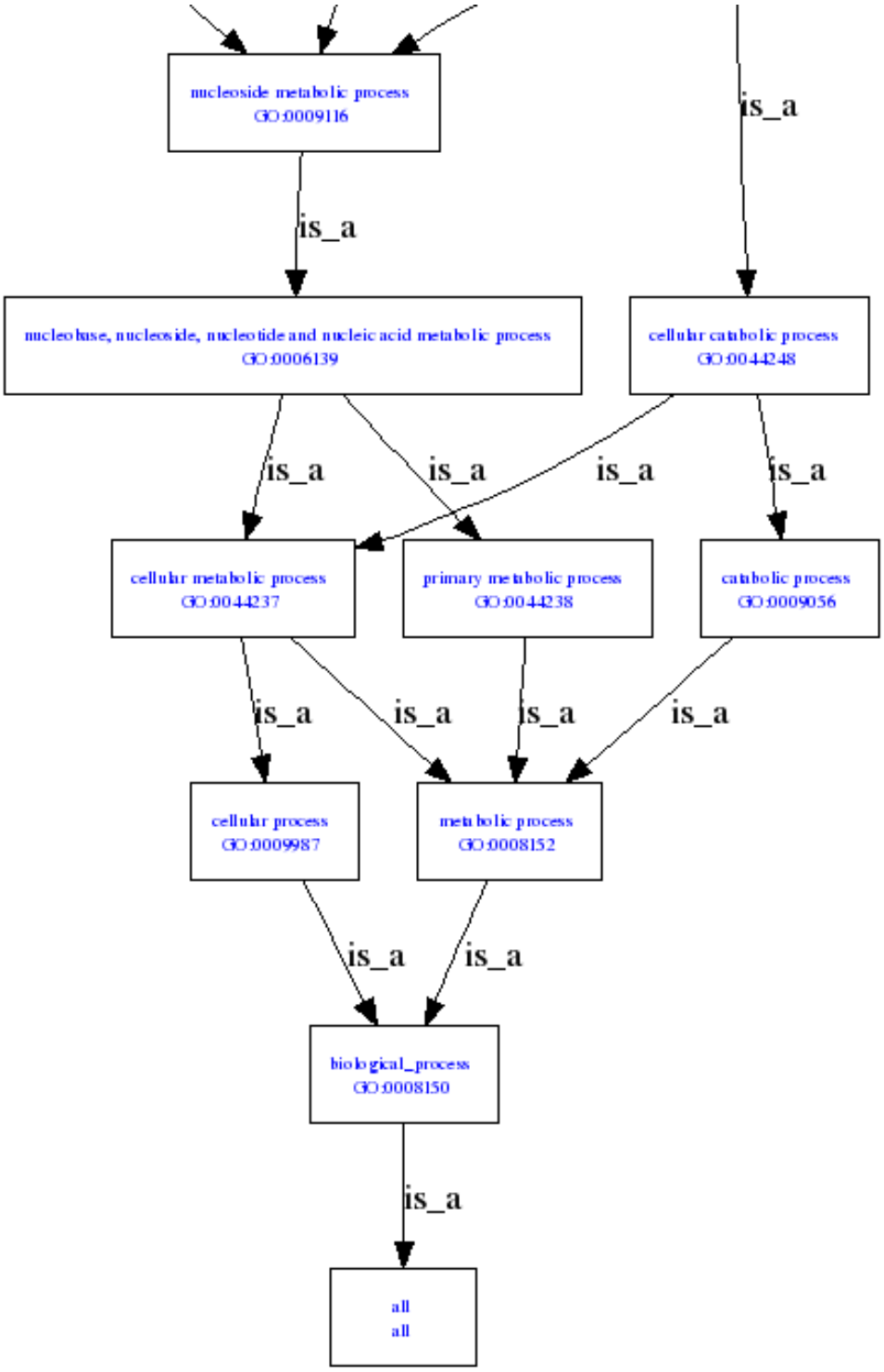
```
LOCUS       NM_172212               2545 bp    mRNA    linear   PRI 25-MAR-2007
DEFINITION  Homo sapiens colony stimulating factor 1 (macrophage) (CSF1),
            transcript variant 4, mRNA.
ACCESSION   NM_172212
VERSION     NM_172212.1  GI:27262666
KEYWORDS    .
SOURCE      Homo sapiens (human)
  ORGANISM  Homo sapiens
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
            Mammalia; Eutheria; Euarchontoglires; Primates; Haplorrhini;
            Catarrhini; Hominidae; Homo.
REFERENCE   1  (bases 1 to 2545)
  AUTHORS   Groblewska,M., Mroczko,B., Wereszczynska-Siemiatkowska,U.,
            Mysliwiec,P., Kedra,B. and Szmitkowski,M.
  TITLE     Serum levels of granulocyte colony-stimulating factor (G-CSF) and
            macrophage colony-stimulating factor (M-CSF) in pancreatic cancer
            patients
  JOURNAL   Clin. Chem. Lab. Med. 45 (1), 30-34 (2007)
   PUBMED   17243911
  REMARK    GeneRIF: significantly higher levels of macrophage-colony
            stimulating factor is associated with pancreatic cancer
[…]
REFERENCE   10 (bases 1 to 2545)
  AUTHORS   Pampfer,S., Tabibzadeh,S., Chuan,F.C. and Pollard,J.W.
  TITLE     Expression of colony-stimulating factor-1 (CSF-1) messenger RNA in
            human endometrial glands during the menstrual cycle: molecular
            cloning of a novel transcript that predicts a cell surface form of
            CSF-1
  JOURNAL   Mol. Endocrinol. 5 (12), 1931-1938 (1991)
   PUBMED   1791839
COMMENT     REVIEWED REFSEQ: This record has been curated by NCBI staff. The
            reference sequence was derived from M27087.1 and BC021117.1.

            Summary: The protein encoded by this gene is a cytokine that
            controls the production, differentiation, and function of
            macrophages. The active form of the protein is found
            extracellularly as a disulfide-linked homodimer, and is thought to
            be produced by proteolytic cleavage of membrane-bound precursors.
            The encoded protein may be involved in development of the placenta.
            Four transcript variants encoding three different isoforms have
            been found for this gene.

            Transcript Variant: This variant (4) differs in the 3' UTR compared
            to variant 1. Both variants 1 and 4 encode isoform a.

            Publication Note:  This RefSeq record includes a subset of the
            publications that are available for this gene. Please see the
            Entrez Gene record to access additional publications.
FEATURES             Location/Qualifiers
     source          1..2545
```

```
                     /organism="Homo sapiens"
                     /mol_type="mRNA"
                     /db_xref="taxon:9606"
                     /chromosome="1"
                     /map="1p21-p13"
     gene            1..2545
                     /gene="CSF1"
                     /note="colony stimulating factor 1 (macrophage); synonyms:
                     MCSF, MGC31930"
                     /db_xref="GeneID:1435"
                     /db_xref="HGNC:2432"
                     /db_xref="HPRD:00388"
                     /db_xref="MIM:120420"
[…]
     STS             714..1777
                     /gene="CSF1"
                     /standard_name="GDB:624803"
                     /db_xref="UniSTS:158360"
ORIGIN
        1 gagggctggc cagtgaggct cggcccgggg aaagtgaaag tttgcctggg tcctctcggc
       61 gccagagccg ctctccgcat cccaggacag cggtgcggcc ctcggccggg gcgcccactc
      121 cgcagcagcc agcgagcgag cgagcgagcg agggcggccg acgcgcccgg ccgggaccca
 […]
     2401 gctggcccac ctcttccccc tcccacccaa gcccacagcc agcccatcag gaagccttcc
     2461 tggcttctcc acaaccttct gactgtcttt tcagtcatgc cccctgctct tttgtatttg
     2521 gctaatagta tatcaatttg cactt
//
```

C.2 Exemples de jeux de données et informations relatives

C.2.1 Gene Ontology




nucleoside metabolic process
GO:0009116
is_a
is_a
nucleobase, nucleoside, nucleotide and nucleic acid metabolic process
GO:0006139
cellular catabolic process
GO:0044248
is_a
is_a
is_a
is_a
cellular metabolic process
GO:0044237
primary metabolic process
GO:0044238
catabolic process
GO:0009056
is_a
is_a
is_a
is_a
cellular process
GO:0009987
metabolic process
GO:0008152
is_a
is_a
biological_process
GO:0008150
is_a
all
all

- all : all [192614]
  - GO:0008150 : biological_process [140750]
    - GO:0009987 : cellular process [71318]
      - GO:0044237 : cellular metabolic process [48370]
        - GO:0044248 : cellular catabolic process [4257]
          - GO:0009164 : nucleoside catabolic process [26]
            - GO:0006152 : purine nucleoside catabolic process [4]
              - GO:0046130 : purine ribonucleoside catabolic process [3]
                - **GO:0006154 : adenosine catabolic process [1]**
            - GO:0042454 : ribonucleoside catabolic process [21]
              - GO:0046130 : purine ribonucleoside catabolic process [3]
                - **GO:0006154 : adenosine catabolic process [1]**
          - GO:0006139 : nucleobase, nucleoside, nucleotide and nucleic acid metabolic process [20817]
            - GO:0009116 : nucleoside metabolic process [216]
              - GO:0009164 : nucleoside catabolic process [26]
                - GO:0006152 : purine nucleoside catabolic process [4]
                  - GO:0046130 : purine ribonucleoside catabolic process [3]
                    - **GO:0006154 : adenosine catabolic process [1]**
                - GO:0042454 : ribonucleoside catabolic process [21]
                  - GO:0046130 : purine ribonucleoside catabolic process [3]
                    - **GO:0006154 : adenosine catabolic process [1]**
              - GO:0042278 : purine nucleoside metabolic process [127]
                - GO:0006152 : purine nucleoside catabolic process [4]
                  - GO:0046130 : purine ribonucleoside catabolic process [3]
                    - **GO:0006154 : adenosine catabolic process [1]**
                - GO:0046128 : purine ribonucleoside metabolic process [123]
                  - GO:0046085 : adenosine metabolic process [9]
                    - **GO:0006154 : adenosine catabolic process [1]**
                  - GO:0046130 : purine ribonucleoside catabolic process [3]
                    - **GO:0006154 : adenosine catabolic process [1]**
              - GO:0009119 : ribonucleoside metabolic process [152]
                - GO:0046128 : purine ribonucleoside metabolic process [123]
                  - GO:0046085 : adenosine metabolic process [9]
                    - **GO:0006154 : adenosine catabolic process [1]**
                  - GO:0046130 : purine ribonucleoside catabolic process [3]
                    - **GO:0006154 : adenosine catabolic process [1]**
                - GO:0042454 : ribonucleoside catabolic process [21]
                  - GO:0046130 : purine ribonucleoside catabolic process [3]
                    - **GO:0006154 : adenosine catabolic process [1]**
    - GO:0008152 : metabolic process [56008]
      - GO:0009056 : catabolic process [4621]
        - GO:0044248 : cellular catabolic process [4257]
          - GO:0009164 : nucleoside catabolic process [26]
            - GO:0006152 : purine nucleoside catabolic process [4]
              - GO:0046130 : purine ribonucleoside catabolic process [3]
                - **GO:0006154 : adenosine catabolic process [1]**
            - GO:0042454 : ribonucleoside catabolic process [21]
              - GO:0046130 : purine ribonucleoside catabolic process [3]
                - **GO:0006154 : adenosine catabolic process [1]**
      - GO:0044237 : cellular metabolic process [48370]
        - GO:0044248 : cellular catabolic process [4257]

- GO:0044248 : cellular catabolic process [4257]
  - GO:0009164 : nucleoside catabolic process [26]
    - GO:0006152 : purine nucleoside catabolic process [4]
      - GO:0046130 : purine ribonucleoside catabolic process [3]
        - **GO:0006154 : adenosine catabolic process [1]**
    - GO:0042454 : ribonucleoside catabolic process [21]
      - GO:0046130 : purine ribonucleoside catabolic process [3]
        - **GO:0006154 : adenosine catabolic process [1]**
- GO:0006139 : nucleobase, nucleoside, nucleotide and nucleic acid metabolic process [20817]
  - GO:0009116 : nucleoside metabolic process [216]
    - GO:0009164 : nucleoside catabolic process [26]
      - GO:0006152 : purine nucleoside catabolic process [4]
        - GO:0046130 : purine ribonucleoside catabolic process [3]
          - **GO:0006154 : adenosine catabolic process [1]**
      - GO:0042454 : ribonucleoside catabolic process [21]
        - GO:0046130 : purine ribonucleoside catabolic process [3]
          - **GO:0006154 : adenosine catabolic process [1]**
    - GO:0042278 : purine nucleoside metabolic process [127]
      - GO:0006152 : purine nucleoside catabolic process [4]
        - GO:0046130 : purine ribonucleoside catabolic process [3]
          - **GO:0006154 : adenosine catabolic process [1]**
      - GO:0046128 : purine ribonucleoside metabolic process [123]
        - GO:0046085 : adenosine metabolic process [9]
          - **GO:0006154 : adenosine catabolic process [1]**
        - GO:0046130 : purine ribonucleoside catabolic process [3]
          - **GO:0006154 : adenosine catabolic process [1]**
    - GO:0009119 : ribonucleoside metabolic process [152]
      - GO:0046128 : purine ribonucleoside metabolic process [123]
        - GO:0046085 : adenosine metabolic process [9]
          - **GO:0006154 : adenosine catabolic process [1]**
        - GO:0046130 : purine ribonucleoside catabolic process [3]
          - **GO:0006154 : adenosine catabolic process [1]**
      - GO:0042454 : ribonucleoside catabolic process [21]
        - GO:0046130 : purine ribonucleoside catabolic process [3]
          - **GO:0006154 : adenosine catabolic process [1]**
- GO:0044238 : primary metabolic process [45701]
  - GO:0006139 : nucleobase, nucleoside, nucleotide and nucleic acid metabolic process [20817]
    - GO:0009116 : nucleoside metabolic process [216]
      - GO:0009164 : nucleoside catabolic process [26]
        - GO:0006152 : purine nucleoside catabolic process [4]
          - GO:0046130 : purine ribonucleoside catabolic process [3]
            - **GO:0006154 : adenosine catabolic process [1]**
        - GO:0042454 : ribonucleoside catabolic process [21]
          - GO:0046130 : purine ribonucleoside catabolic process [3]
            - **GO:0006154 : adenosine catabolic process [1]**
      - GO:0042278 : purine nucleoside metabolic process [127]
        - GO:0006152 : purine nucleoside catabolic process [4]
          - GO:0046130 : purine ribonucleoside catabolic process [3]
            - **GO:0006154 : adenosine catabolic process [1]**
        - GO:0046128 : purine ribonucleoside metabolic process [123]
          - GO:0046085 : adenosine metabolic process [9]
            - **GO:0006154 : adenosine catabolic process [1]**
          - GO:0046130 : purine ribonucleoside catabolic process [3]
            - **GO:0006154 : adenosine catabolic process [1]**
      - GO:0009119 : ribonucleoside metabolic process [152]
        - GO:0046128 : purine ribonucleoside metabolic process [123]
          - GO:0046085 : adenosine metabolic process [9]
            - **GO:0006154 : adenosine catabolic process [1]**
          - GO:0046130 : purine ribonucleoside catabolic process [3]
            - **GO:0006154 : adenosine catabolic process [1]**
        - GO:0042454 : ribonucleoside catabolic process [21]
          - GO:0046130 : purine ribonucleoside catabolic process [3]
            - **GO:0006154 : adenosine catabolic process [1]**

C.2.2 UMLS

C.2.2.1 Relations sémantiques

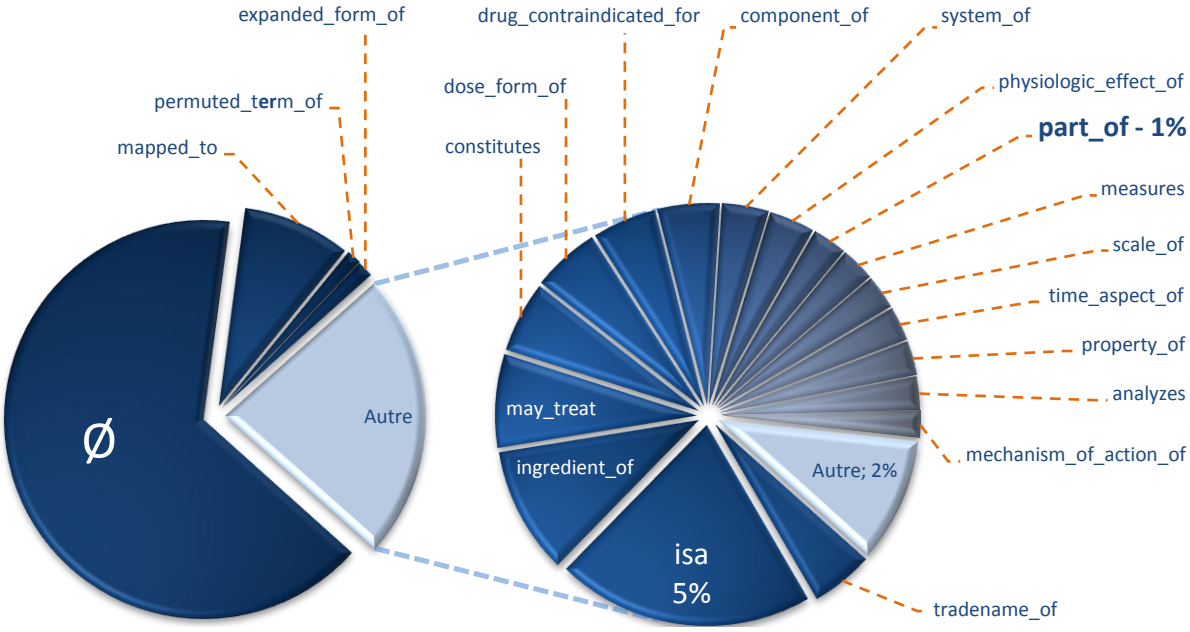


Figure C.12 – Distribution des quantités correspondant à chaque type de relation sémantique dans UMLS.

Relations	Redondances			
Ø		4346783	65,5%	
mapped_to	mapped_from	580020	8,7%	
permuted_term_of	has_permuted_term	79493	1,2%	
expanded_form_of	has_expanded_form	88263	1,3%	
tradename_of	has_tradename	77142	1,2%	
isa	inverse_isa / sib_in_isa	319381	4,8%	21,7%
ingredient_of	has_ingredient	156936	2,4%	10,7%
may_treat	may_be_treated_by	115055	1,7%	7,8%
constitutes	consists_of	87795	1,3%	6,0%
dose_form_of	has_dose_form	83648	1,3%	5,7%
drug_contraindicated_for	has_contraindicated_drug	79335	1,2%	5,4%
component_of	has_component	75135	1,1%	5,1%
system_of	has_system	56442	0,9%	3,8%
physiologic_effect_of	has_physiologic_effect	56219	0,8%	3,8%
part_of	has_part / sib_in_part_of	42637	0,6%	2,9%
measures	measured_by	42549	0,6%	2,9%
scale_of	has_scale	42400	0,6%	2,9%
time_aspect_of	has_time_aspect	42259	0,6%	2,9%
property_of	has_property	42232	0,6%	2,9%
Analyzes	analyzed_by	40735	0,6%	2,8%
mechanism_of_action_of	has_mechanism_of_action	32070	0,5%	2,2%
classified_as	classifies	23410	0,4%	1,6%

Relations	Redondances			
method_of	has_method	21379	0,3%	1,5%
suffix_of	has_suffix	20487	0,3%	1,4%
may_prevent	may_be_prevented_by	18805	0,3%	1,3%
branch_of	has_branch / sib_in_branch_of	10392	0,2%	0,7%
Use	used_for	10384	0,2%	0,7%
challenge_of	has_challenge	8315	0,1%	0,6%
contraindicated_with	has_contraindication	6218	0,1%	0,4%
Induces	induced_by	3854	0,1%	0,3%
tributary_of	has_tributary / sib_in_tributary_of	3280		0,2%
See	see_from	3191		0,2%
may_diagnose	may_be_diagnosed_by	3037		0,2%
location_of	has_location	2882		0,2%
supersystem_of	has_supersystem	2462		0,2%
pharmacokinetics_of	has_pharmacokinetics	2237		0,2%
divisor_of	has_divisor	2136		0,1%
clinically_similar		1952		0,1%
encodes_gene_product	encoded_by_gene	1850		0,1%
metabolizes	metabolized_by	1761		0,1%
form_of	has_form	1597		0,1%
mth_expanded_form_of	mth_has_expanded_form	1298		0,1%
site_of_metabolism	metabolic_site_of	525		
has_multi_level_category	has_single_level_category	491		
Exhibits	exhibited_by	476		
precise_ingredient_of	has_precise_ingredient	333		
outcome_of	has_outcome	258		
manifestation_of	has_manifestation	235		
evaluation_of	has_evaluation	105		
adjustment_of	has_adjustment	68		
plain_text_form_of	has_plain_text_form	51		
version_of	has_version	46		
Uses	used_by	21		
associated_disease	associated_genetic_condition	16		
mth_plain_text_form_of	mth_has_plain_text_form	12		
degree_of	has_degree	10		
diagnoses	diagnosed_by	9		
conceptual_part_of	has_conceptual_part	5		
associated_with		2		
translation_of	has_translation	4		
larger_than	smaller_than	3		
treats	treated_by	2		
result_of	has_result	1		

Figure C.13 – Détail de chaque type de relation sémantique d'UMLS avec le nombre d'instance dans l'entrepôt.

C.2.2.2 Sources

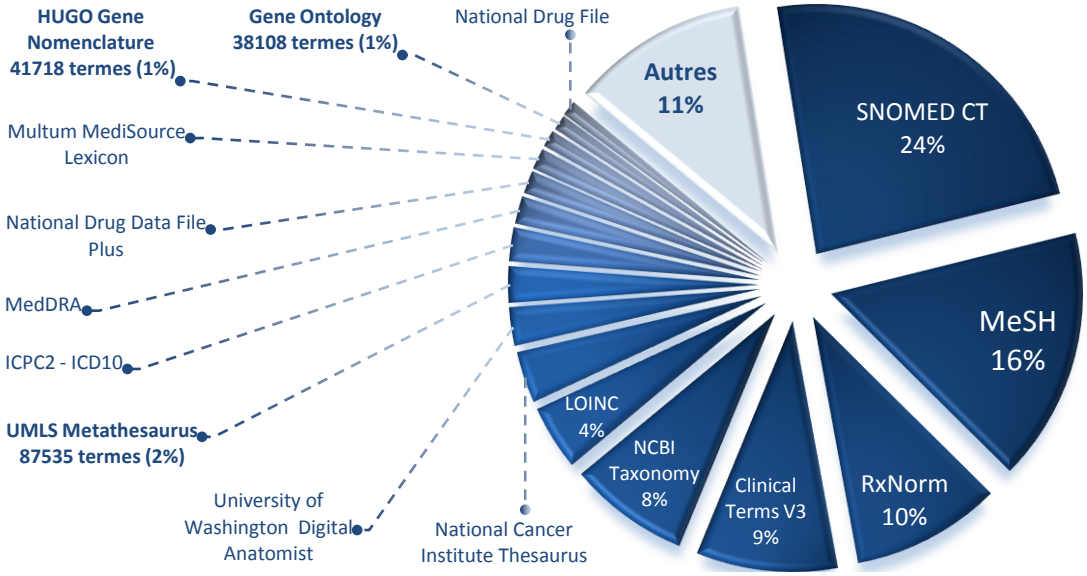


Figure C.14 – Distribution des données dans UMLS en fonction des sources.

Resource	Nombre de termes	% d'UMLS
SNOMED CT	902718	23,61%
MeSH	624557	16,34%
RxNorm	369461	9,66%
Clinical Terms V3	347312	9,08%
NCBI Taxonomy	297559	7,78%
LOINC	158843	4,15%
National Cancer Institute Thesaurus	126342	3,30%
University of Washington Digital Anatomist	92913	2,43%
UMLS Metathesaurus	87535	2,29%
ICPC2 - ICD10	81799	2,14%
MedDRA	67837	1,77%
National Drug Data File Plus	56640	1,48%
Multum MediSource Lexicon	52575	1,38%
HUGO Gene Nomenclature	41718	1,09%
Gene Ontology	38108	1,00%
National Drug File	37919	0,99%
Autres	439127	11,49%

Figure C.15 – Détail de la contribution dans UMLS de chaque source en nombre de termes.

C.2.2.3 UMLS Browser

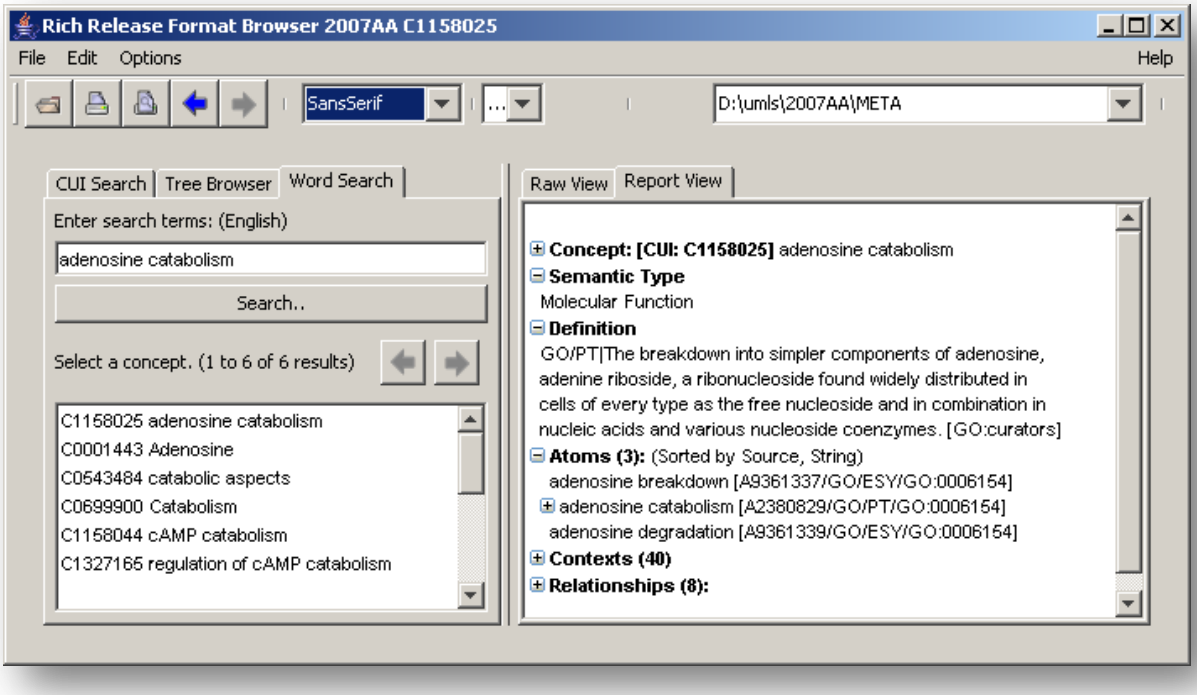


Figure C.16 – Client graphique en Java permettant d'interroger UMLS localement.

C.2.2.4 UMLSKS

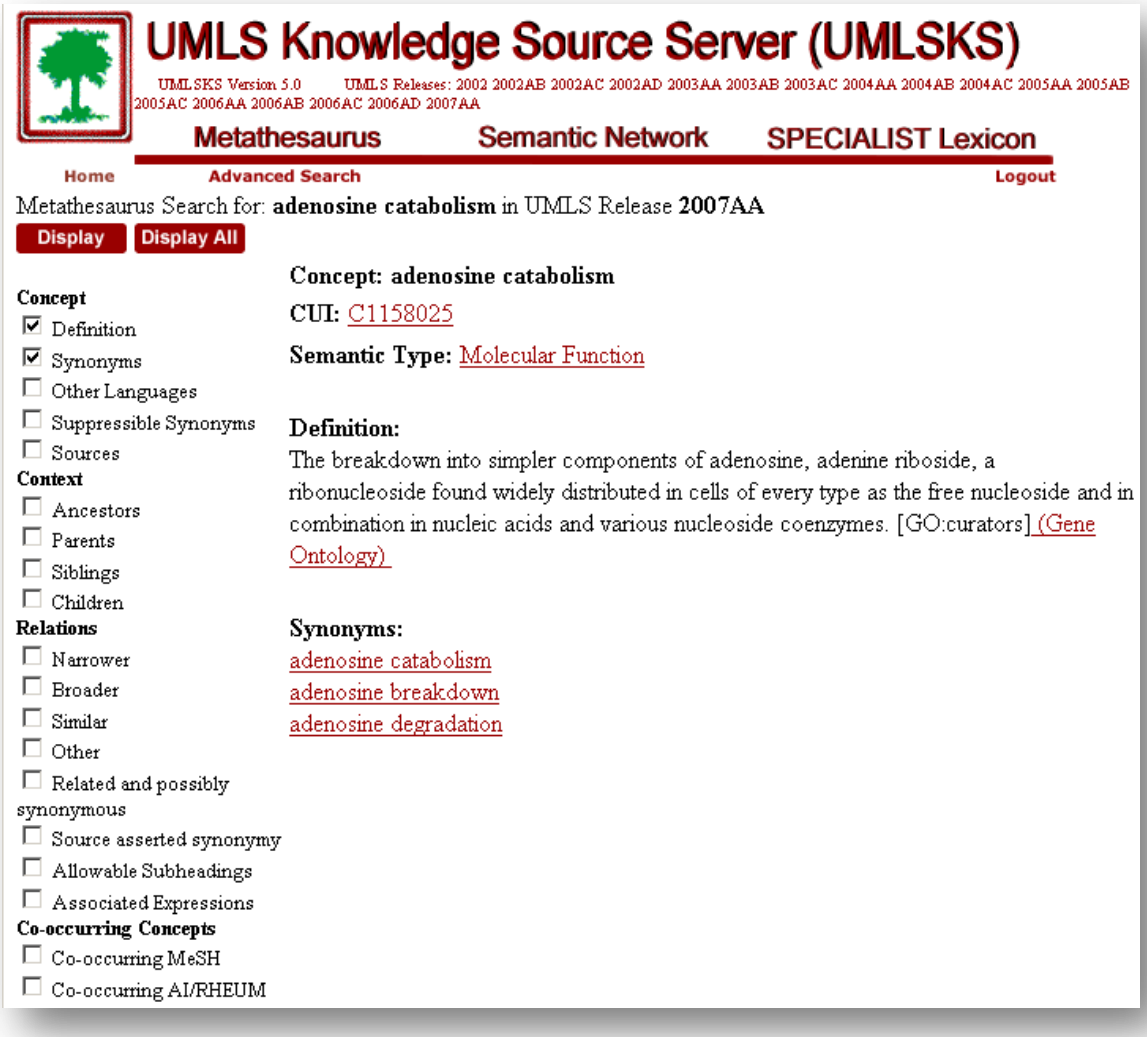


Figure C.17 – UMLSKS est le portail d'accès en ligne aux données d'UMLS.

C.1 Captures de logiciels et portails

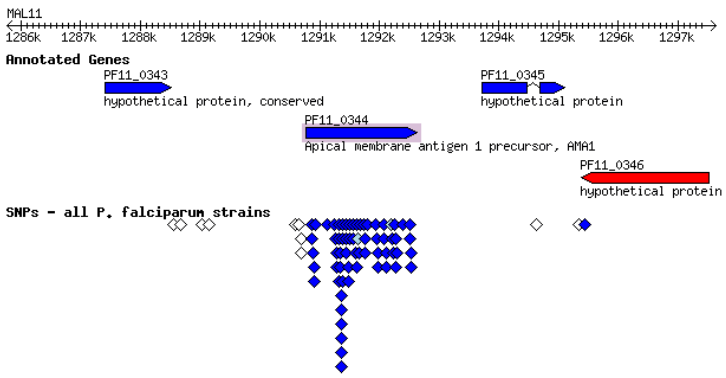
C.1.1 PlasmoDB (GUS)

PF11_0344
Apical membrane antigen 1 precursor, AMA1

P. falciparum 3D7 protein coding gene on MAL11 from 1290767 to 1292635 (1868 bp) Rele

Annotation ▸ Protein ▸ Expression ▸ Sequence ▸ PF11_0344 in PlasmoDB 4.4 ▸

Genomic Context Hide [Data Sourc



View this sequence in the genome browser
(*use right click or ctrl-click to open in a new window*)

SNPs Summary Show [Data Sourc

Annotation back to

Paralogs and Plasmodium Orthologs Hide [Data Sourc

Find PF11_0344 in the OrthoMCL database

Gene	Species	Product
PB000821.01.0	Plasmodium berghei	apical membrane antigen 1 precursor, putative
PC300445.00.0	Plasmodium chabaudi	hypothetical protein
PC301665.00.0	Plasmodium chabaudi	apical membrane antigen 1 precursor, putative
Pv092275	Plasmodium vivax Sal-1	apical merozoite antigen 1
PY01581	Plasmodium yoelii yoelii str. 17XNL	apical membrane antigen-1

EC Number *none* [Data Sourc

GO Terms Show [Data Sourc

Aliases Hide [Data Sourc

Alias
1399.t00180

Notes Show [Data Sourc

Publications View

Metabolic Pathways Show [Data Sourc

PlasmoCyc View

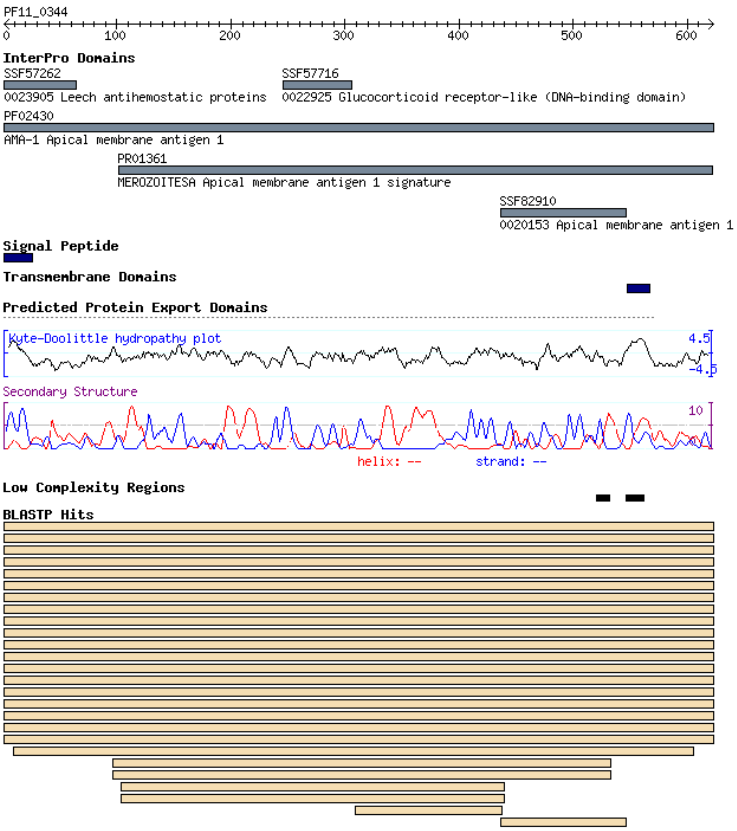
MR4 Reagents Show [Data Sourc

User Comments Show

Add a comment on PF11_0344

Protein back to

Protein Features Hide [Data Sourc



Y2H Interactions *none* [Data Sourc

Molecular Weight: 72042 Da

Isoelectric Point: 5.23

Mass Spec.-based Expression Evidence Hide [Data Sourc

Lifecycle Stage	Algorithm	Coverage	Spans	Sequences	Spectra
merozoite	Sequest	14.1	5	5	45
sporozoite	Sequest	14.5	5	5	12

Protein Linkouts Show

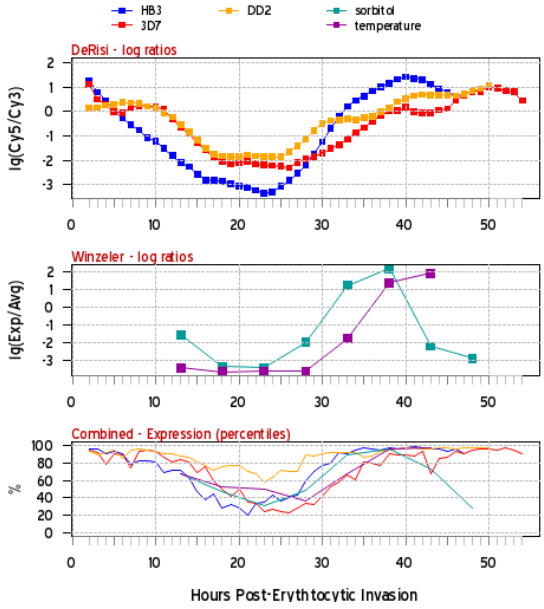
Similarities to Protein Data Bank (PDB) Chains Show [Data Sourc

3D Structure Predictions Show [Data Sourc

Expression back to

Mapped Array Elements Show [Data Sourc

Overlay of Intraerythrocytic Expression Profiles Hide [Data Sourc



Studies by the Derisi Lab of *P. falciparum* strains HB3, 3D7, and Dd2 used glass slide arrays.

Studies by the Winzeler Lab of Sorbitol- and Temperature-synchronized parasites (of the same three strains) used Affymetrix oligonucleotide arrays.

More on mapping time points between time courses

Intraerythrocytic 3D7 (photolithographic oligo array) Show [Data Sourc

Gametocyte 3D7/NF54 (photolithographic oligo array) Show [Data Sourc

Developmental series 3D7 (glass slide oligo array) Show [Data Sourc

Developmental series Dd2 (glass slide oligo array) Show [Data Sourc

Developmental series HB3 (glass slide oligo array) Show [Data Sourc

Sequence back to

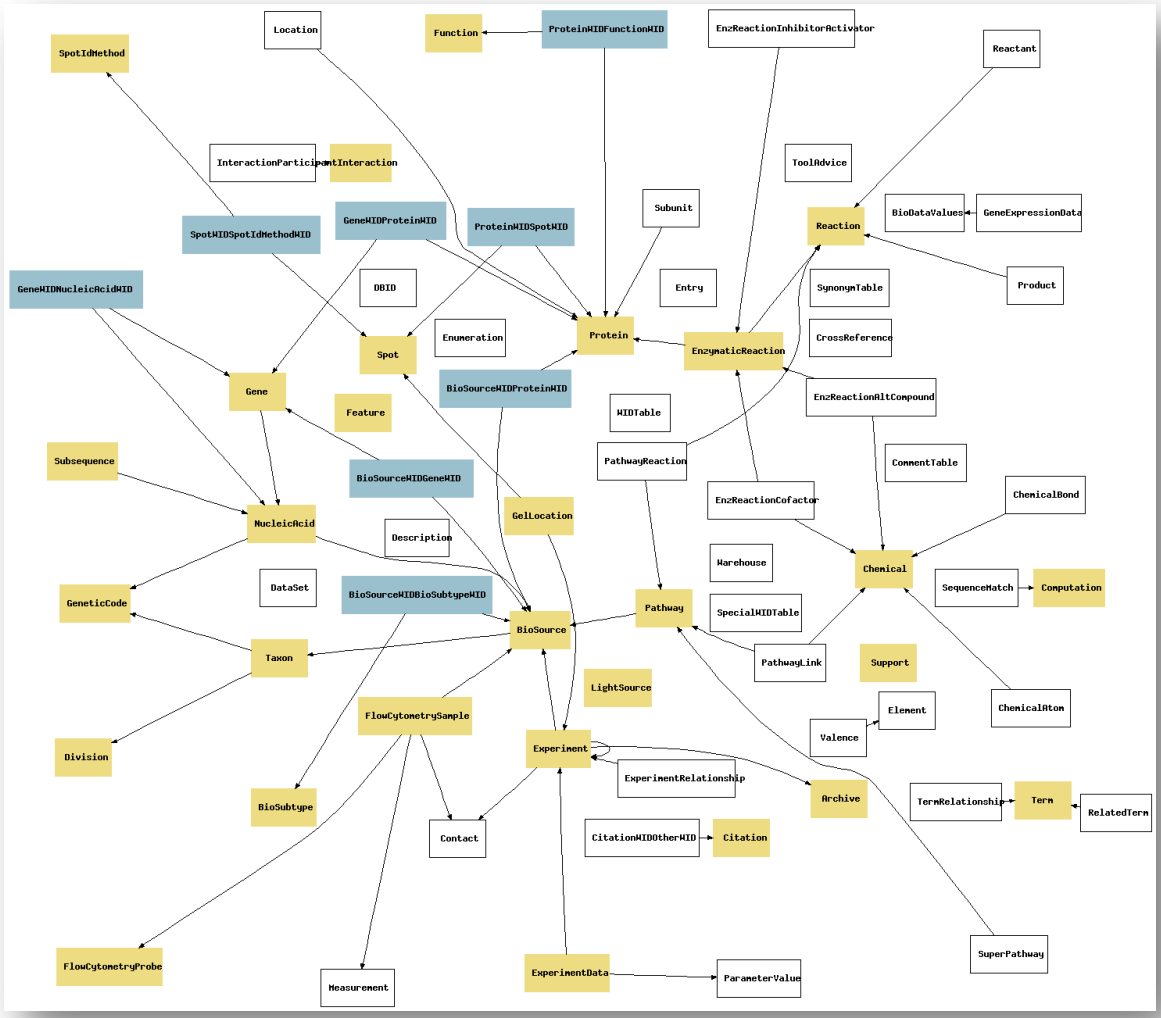
Predicted Protein Sequence Show

Predicted RNA/mRNA Sequence (introns spliced out) Show

Coding Sequence Show

C.1.2 BioWarehouse - Schéma

Le schéma relationnel est composé de 179 tables (relations) dont 87 sont des entités et 59 sont produites par des associations *« many-to-many ».* Le graphique ci-dessous ne représente qu'une partie du schéma (29 entités).



C.1.3 GenoLink

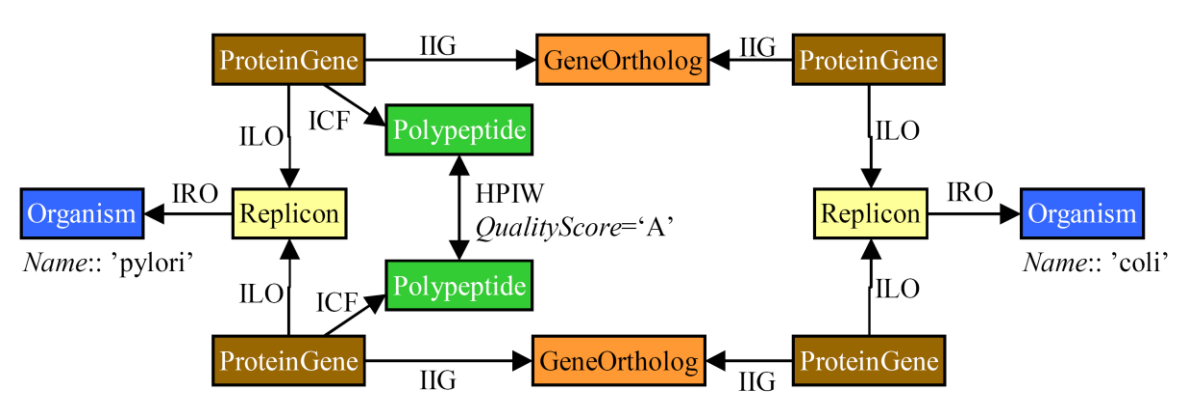


Figure C.18 – A partir d'un organisme (« pylori »), on cherche à inférer automatiquement des interactions (de qualité « A ») entre deux protéines similaires dans un autre organisme (« coli »). Des gènes sont orthologues s'ils sont similaires et partagent un ancêtre commun.

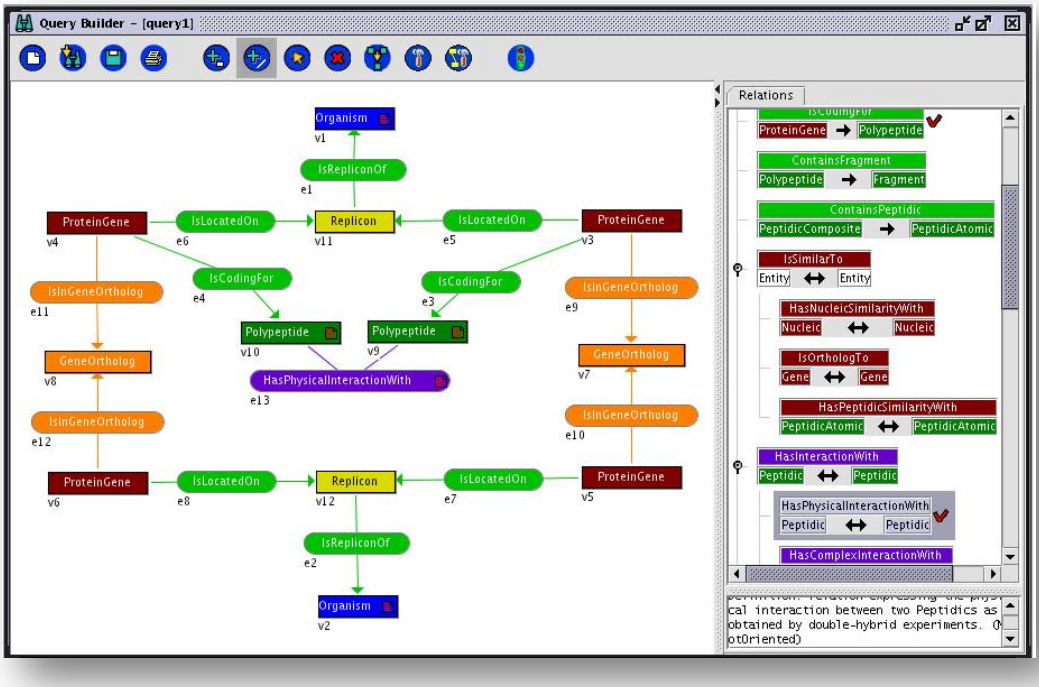


Figure C.19 – Cette fenêtre montre l'exemple d'une requête dans GenoLink. La partie gauche contient la requête. Les boîtes carrées (foncées) sont des entités, les boîtes arrondies (plus claires) représentent les étiquettes des relations. La partie droite propose une vue du métamodèle pour construire la requête.
A partir d'une interaction observée entre deux protéines dans un organisme (v1, en haut du graphe), on cherche à inférer des interactions entre deux protéines similaires dans un autre organisme (orthologues).

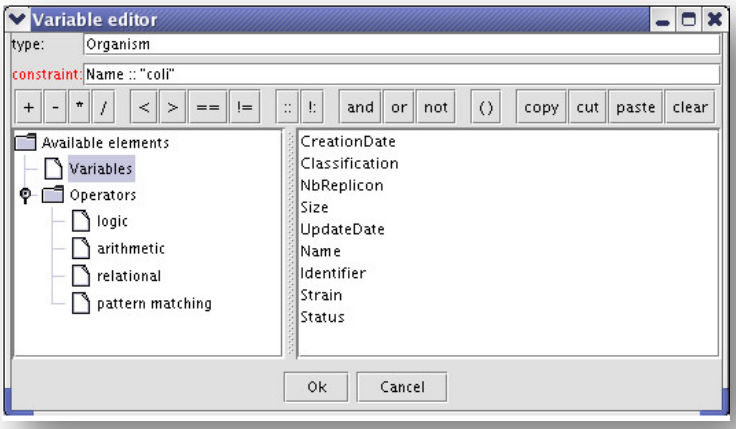


Figure C.20 – Afin de préciser des conditions de filtrage (nom d'un organisme, etc.), cette boite de dialogue permet d'éditer les conditions rattachées à chaque entité et relation.

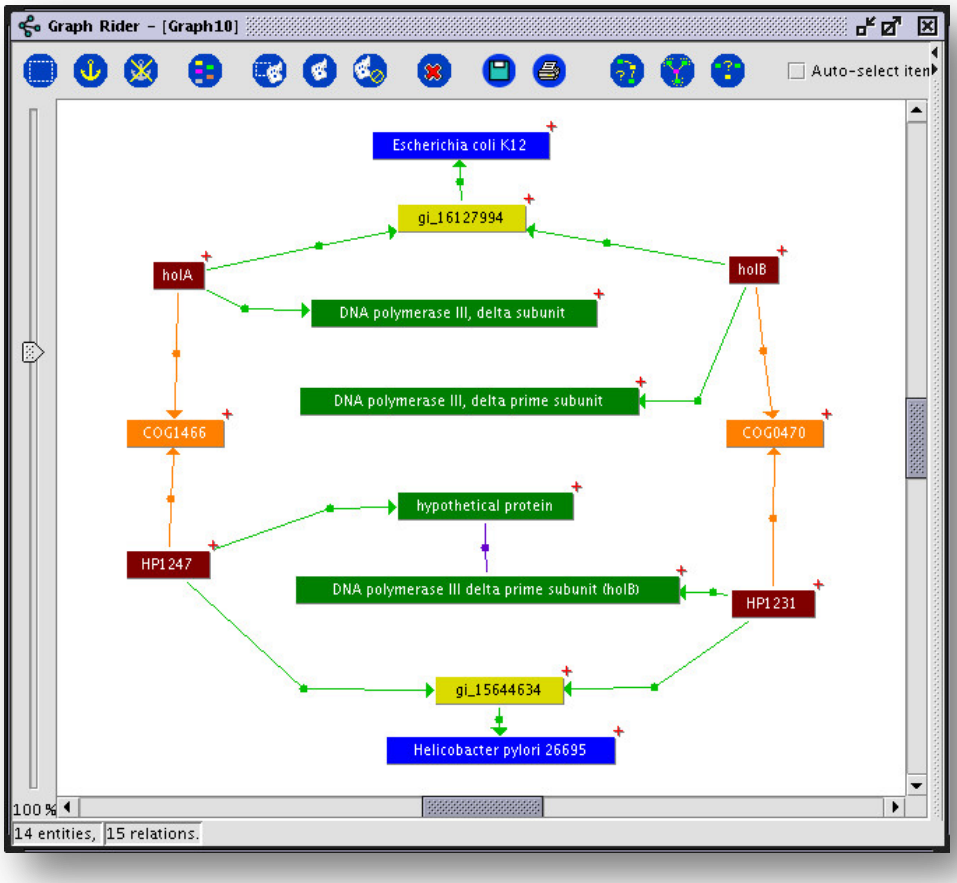
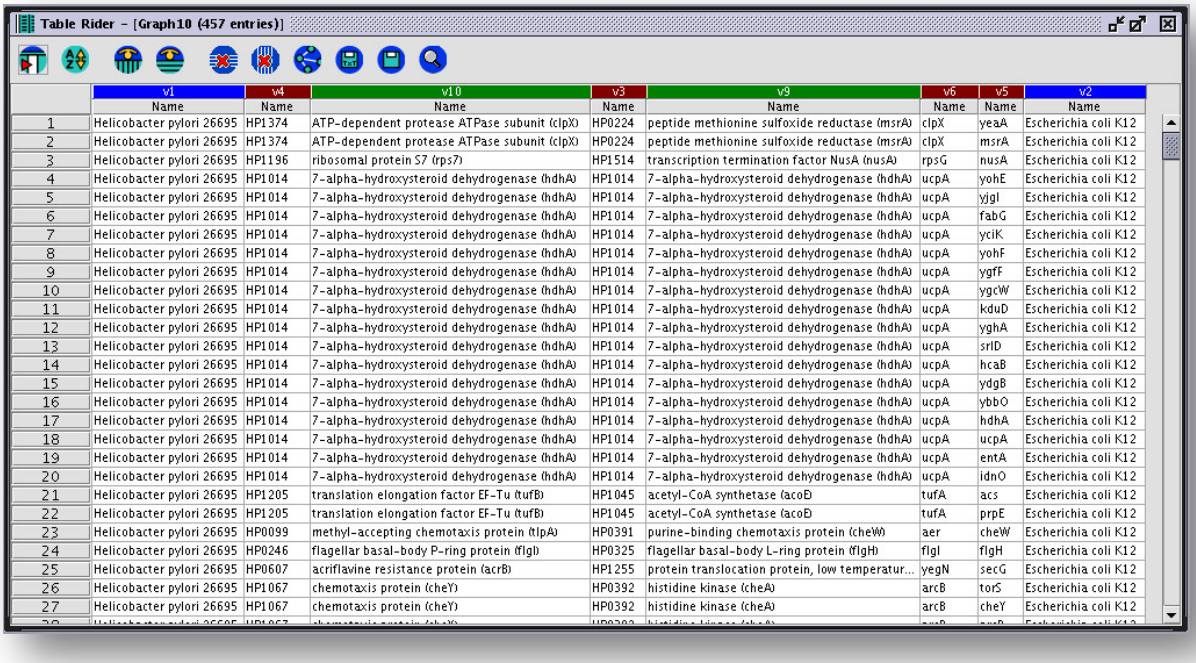


Figure C.21 – Voici une capture d'une instance de résultat correspondant à la requête précédente, visualisée sous forme de graphe. Des interactions permettent d'obtenir des informations sur les différentes entités, leur sources, leur identifiants, etc.



	v1	v4	v10	v3	v9	v6	v5	v2
	Name	Name	Name	Name	Name	Name	Name	Name
1	Helicobacter pylori 26695	HP1374	ATP-dependent protease ATPase subunit (clpX)	HP0224	peptide methionine sulfoxide reductase (msrA)	clpX	yeaA	Escherichia coli K12
2	Helicobacter pylori 26695	HP1374	ATP-dependent protease ATPase subunit (clpX)	HP0224	peptide methionine sulfoxide reductase (msrA)	clpX	msrA	Escherichia coli K12
3	Helicobacter pylori 26695	HP1196	ribosomal protein S7 (rps7)	HP1514	transcription termination factor NusA (nusA)	rpsG	nusA	Escherichia coli K12
4	Helicobacter pylori 26695	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	ucpA	yohE	Escherichia coli K12
5	Helicobacter pylori 26695	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	ucpA	yjgI	Escherichia coli K12
6	Helicobacter pylori 26695	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	ucpA	fabG	Escherichia coli K12
7	Helicobacter pylori 26695	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	ucpA	yciK	Escherichia coli K12
8	Helicobacter pylori 26695	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	ucpA	yohF	Escherichia coli K12
9	Helicobacter pylori 26695	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	ucpA	ygfF	Escherichia coli K12
10	Helicobacter pylori 26695	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	ucpA	ygcW	Escherichia coli K12
11	Helicobacter pylori 26695	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	ucpA	kduD	Escherichia coli K12
12	Helicobacter pylori 26695	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	ucpA	yghA	Escherichia coli K12
13	Helicobacter pylori 26695	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	ucpA	srlD	Escherichia coli K12
14	Helicobacter pylori 26695	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	ucpA	hcaB	Escherichia coli K12
15	Helicobacter pylori 26695	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	ucpA	ydgB	Escherichia coli K12
16	Helicobacter pylori 26695	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	ucpA	ybbO	Escherichia coli K12
17	Helicobacter pylori 26695	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	ucpA	hdhA	Escherichia coli K12
18	Helicobacter pylori 26695	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	ucpA	ucpA	Escherichia coli K12
19	Helicobacter pylori 26695	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	ucpA	entA	Escherichia coli K12
20	Helicobacter pylori 26695	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	HP1014	7-alpha-hydroxysteroid dehydrogenase (hdhA)	ucpA	idnO	Escherichia coli K12
21	Helicobacter pylori 26695	HP1205	translation elongation factor EF-Tu (tufB)	HP1045	acetyl-CoA synthetase (acoD	tufA	acs	Escherichia coli K12
22	Helicobacter pylori 26695	HP1205	translation elongation factor EF-Tu (tufB)	HP1045	acetyl-CoA synthetase (acoD	tufA	prpE	Escherichia coli K12
23	Helicobacter pylori 26695	HP0099	methyl-accepting chemotaxis protein (tlpA)	HP0391	purine-binding chemotaxis protein (cheW)	aer	cheW	Escherichia coli K12
24	Helicobacter pylori 26695	HP0246	flagellar basal-body P-ring protein (flgI)	HP0325	flagellar basal-body L-ring protein (flgH)	flgI	flgH	Escherichia coli K12
25	Helicobacter pylori 26695	HP0607	acriflavine resistance protein (acrB)	HP1255	protein translocation protein, low temperatur...	yegN	secG	Escherichia coli K12
26	Helicobacter pylori 26695	HP1067	chemotaxis protein (cheY)	HP0392	histidine kinase (cheA)	arcB	torS	Escherichia coli K12
27	Helicobacter pylori 26695	HP1067	chemotaxis protein (cheY)	HP0392	histidine kinase (cheA)	arcB	cheY	Escherichia coli K12

Figure C.22 – Le résultat peut aussi être visualisé de façon tabulaire.

C.1.4 Entrez

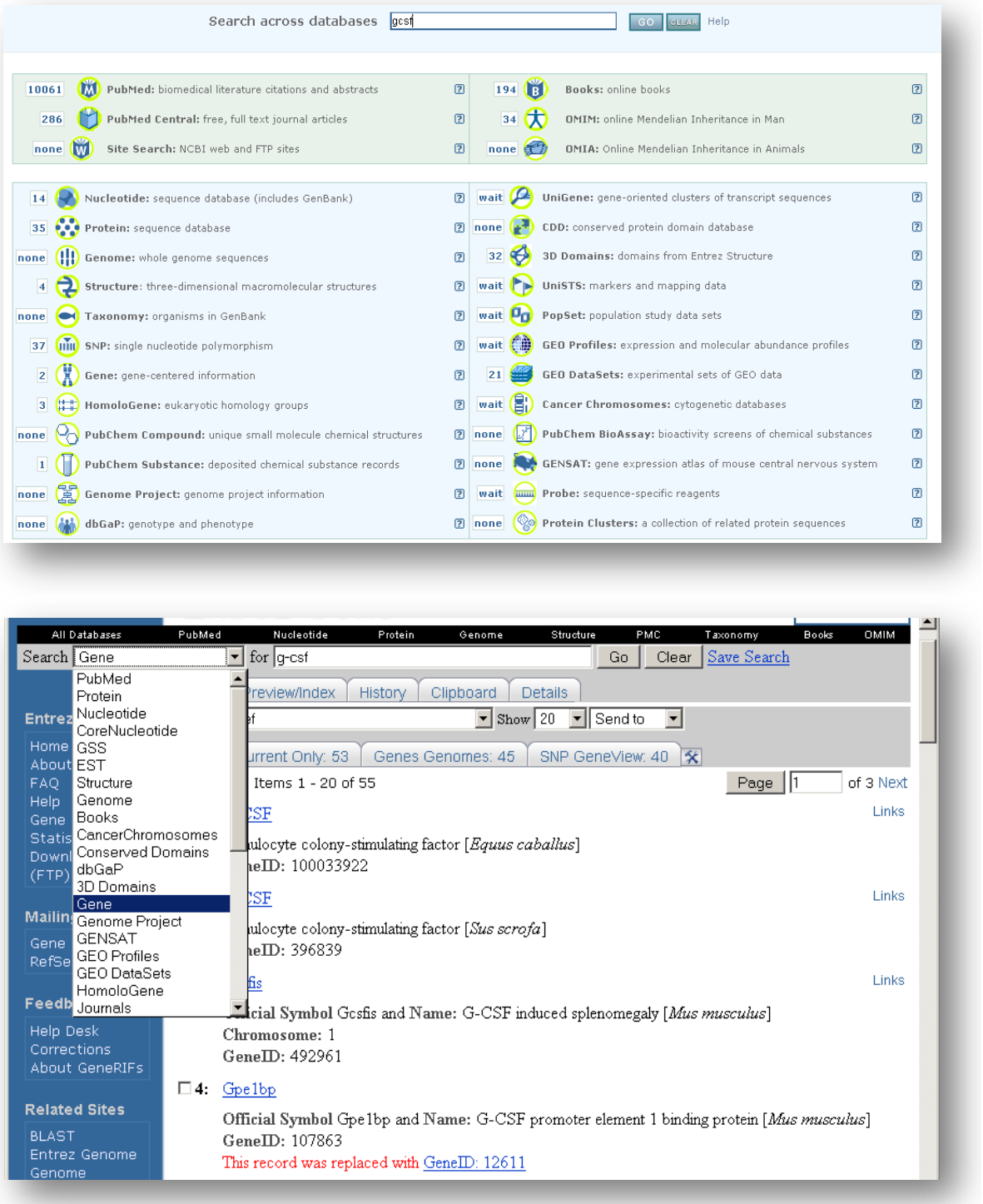
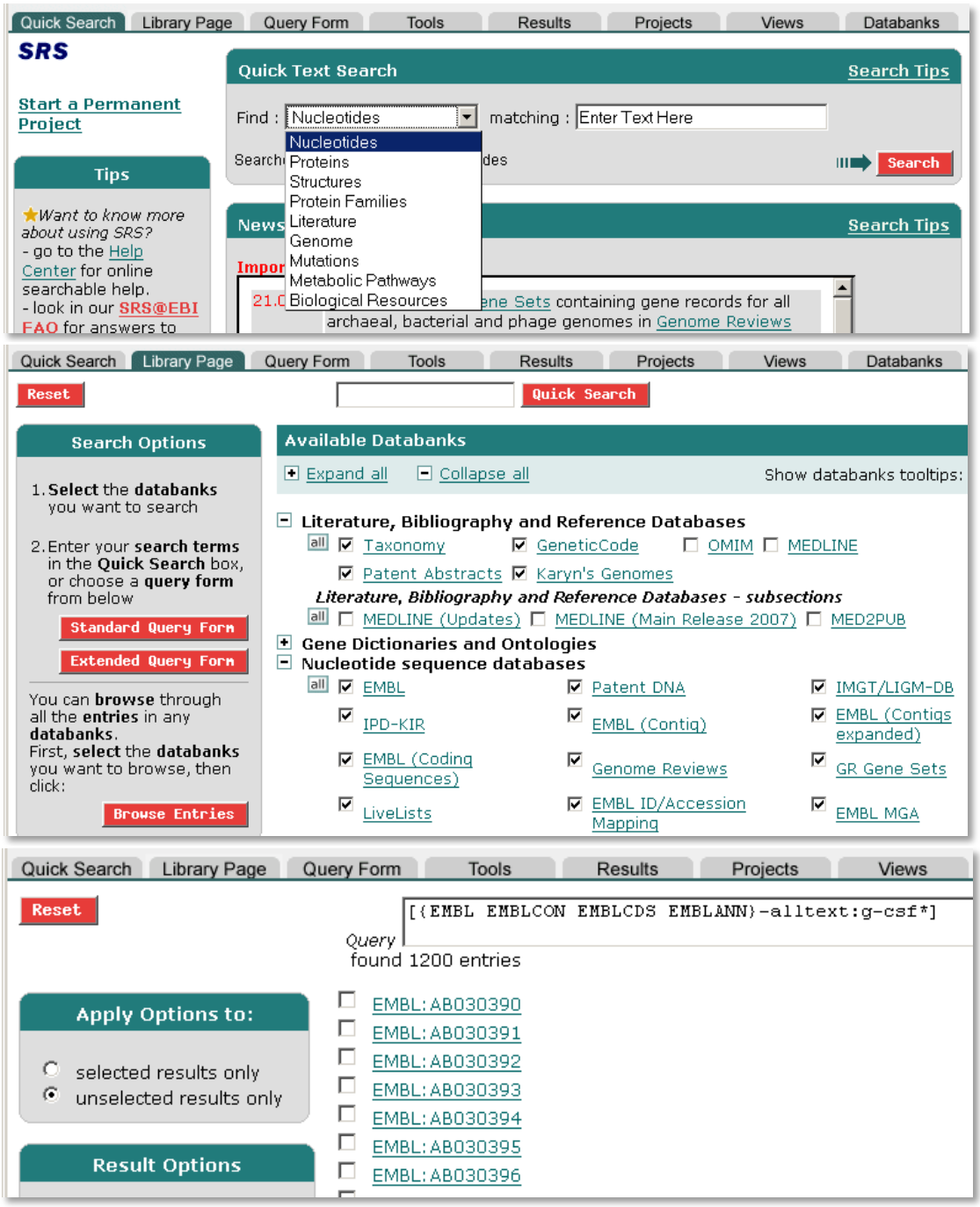


Figure C.23 – Captures d'écran du portail Entrez du NCBI. Les sources sont interrogées en parallèle. Petit à petit, les réponses sont collectées (« wait » indique l'attente de réponse pour une source).

C.1.5 SRS@EBI


Quick Search
Library Page
Query Form
Tools
Results
Projects
Views
Databanks
SRS
Start a Permanent Project
Quick Text Search
Search Tips
Find : Nucleotides
matching : Enter Text Here
Nucleotides
Proteins
Structures
Protein Families
Literature
Genome
Mutations
Metabolic Pathways
Biological Resources
Search
Tips
Want to know more about using SRS?
- go to the Help Center for online searchable help.
- look in our SRS@EBI FAQ for answers to
Search Tips
Gene Sets containing gene records for all archaeal, bacterial and phage genomes in Genome Reviews
Reset
Quick Search
Search Options
1. Select the databanks you want to search
2. Enter your search terms in the Quick Search box, or choose a query form from below
Standard Query Form
Extended Query Form
You can browse through all the entries in any databanks.
First, select the databanks you want to browse, then click:
Browse Entries
Available Databanks
Expand all
Collapse all
Show databanks tooltips:
Literature, Bibliography and Reference Databases
Taxonomy
GeneticCode
OMIM
MEDLINE
Patent Abstracts
Karyn's Genomes
Literature, Bibliography and Reference Databases - subsections
MEDLINE (Updates)
MEDLINE (Main Release 2007)
MED2PUB
Gene Dictionaries and Ontologies
Nucleotide sequence databases
EMBL
Patent DNA
IMGT/LIGM-DB
IPD-KIR
EMBL (Contig)
EMBL (Contigs expanded)
EMBL (Coding Sequences)
Genome Reviews
GR Gene Sets
LiveLists
EMBL ID/Accession Mapping
EMBL MGA
Reset
[{EMBL EMBLCON EMBLCDS EMBLANN}-alltext:g-csf*]
Query
found 1200 entries
Apply Options to:
selected results only
unselected results only
Result Options
EMBL:AB030390
EMBL:AB030391
EMBL:AB030392
EMBL:AB030393
EMBL:AB030394
EMBL:AB030395
EMBL:AB030396

Cartographie des connaissances : l'intégration et la visualisation au service de la biologie. Application à l'ingénierie des connaissances et à l'analyse de données d'expression de gènes.

Résumé : Ce mémoire s'inscrit dans un axe stratégique du groupement des Ecoles des Mines : GEMBIO. Dans ce contexte, plusieurs collaborations ont été initiées, notamment avec des chercheurs de l'Institut Pasteur de Paris, de l'Inserm/Hôpitaux de Paris, et du CEA dans le cadre du programme ToxNuc-e. De ces échanges, est née notre problématique. Plus d'un millier de bases de données biologiques sont disponibles en ligne. Leur exploitation et le croisement de leurs contenus entraînent souvent ce constat des chercheurs biologistes : « J'ai souvent une vingtaine de fenêtres ouvertes sur mon écran : je m'y perds ». Souvent l'analyse et le croisement des données est fait par simple copier-coller dans un tableur.

Si l'intégration de données à apporté des solutions ponctuelles à des problèmes particuliers, elle ne propose pas pour autant une réponse concrète à la multiplicité des fenêtres pour l'utilisateur, à la surcharge d'information, et à la difficulté de croiser l'information provenant de plusieurs sources hétérogènes.

Nous proposons un environnement de cartographie des connaissances biologiques qui facilite l'intégration et la visualisation des données biologiques. Basé sur un métamodèle simple de graphe, I²DEE (Integrated and Interactive Data Exploration Environment) se veut souple et extensible afin de répondre aux besoins des différentes approches existantes de l'intégration. Il permet un accès homogène aux principales ressources biologiques et son adaptabilité offre des réponses visuelles personnalisées à des tâches spécifiques.

Après une analyse des besoins des chercheurs biologistes et l'identification des problématiques de traitement de l'information sous-jacentes, un état de l'art de l'intégration de données hétérogènes est présenté. L'approche proposée reprend les principes existants en architecture des IHM et en cartographie géographique. L'environnement I2DEE est alors présenté à partir de son architecture et son métamodèle. Deux modules de l'environnement sont détaillés : l'entrepôt de données biologiques et la boîte à outils graphique permettant de construire rapidement des applications adaptées. Des résultats ont été obtenus dans deux contextes applicatifs distincts : l'ingénierie terminologique et ontologique, et l'analyse de données d'expression de gènes issues de puces à ADN. Ils sont discutés et analysés en regard des objectifs initialement fixés.

Mots-clés : Intégration de données biologiques, visualisation, ingénierie des connaissances, cartographie des connaissances.

Knowledge mapping: biological data integration and visualization applied to knowledge engineering and gene expression data analysis.

Abstract: The biomedical domain uses more than 1000 online databases. Crossing and analyzing their content is hard, and users often report the following kind of comment: "There are frequently about 20 windows on my desktop; so I'm lost". Data crossing and analysis is also mostly done by manually copying and pasting data into a spreadsheet. Data integration community advances do not concretely address the user's needs for better visualization and integration tools.

This thesis proposes a biological knowledge mapping environment that simplifies integration and visualization of biological data. I²DEE (an Integrated and Interactive Data Exploration Environment) is based upon a simple graph metamodel. This metamodel confers I²DEE flexibility and extensibility for interoperating with existing data integration approaches. I²DEE provides a visual and homogeneous biological data access and can adapt to specialized user tasks. To demonstrate I²DEE versatility, two applications have been experimented in the context of microarray gene expression data analysis and knowledge engineering.

Keywords: Biological data integration, visualization, knowledge engineering, knowledge mapping.

Discipline : Informatique

Laboratoire de rattachement : Centre de Recherche LGI2P de l'école des Mines d'Alès.