N° d'ordre : 55 Année 2012

# UNIVERSITE ABDELMALEK ESSAADI
## FACULTE DES SCIENCES et TECHNIQUES
## TANGER

UFR : Valorisation Biotechnologique des Micro-Organismes

# THESE

**Présentée**

**Pour l'obtention du**

**DOCTORAT EN SCIENCES ET TECHNIQUES**

**Par :**
**Kamar MARRAKCHI**

**Discipline : Bioinformatique**

**Spécialité : Bioinformatique**

**Une approche hybride pour une intégration sémantique des données biologiques de Pseudomonas**

**Soutenue le 19/12/2012 devant le Jury**

| | | |
|---|---|---|
| **Pr. Ahmed LAMARTI** | **Faculté des Sciences– Tétouan** | **Président** |
| **Pr. Mohamed ETTAYEBI** | **Faculté des Sciences – Fès** | **Rapporteur** |
| **Pr. Ismael Navas DELGADO** | **E.T.S. Ingeniería Informática – Málaga** | **Rapporteur** |
| **Pr. M'hamed AIT KBIR** | **Faculté des Sciences et Techniques – Tanger** | **Examinateur** |
| **Pr. Khalid LAIRINI** | **Faculté des Sciences et Techniques – Tanger.** | **Examinateur** |
| **Pr. José F. Aldana MONTES** | **E.T.S. Ingeniería Informática – Málaga** | **Co-Directeur** |
| **Pr. Badr Din ROSSI HASSANI** | **Faculté des Sciences et Techniques – Tanger.** | **Co-Directeur** |

# Une approche hybride pour une intégration sémantique des données biologiques de Pseudomonas

# Remerciement

# Résumé

Les Pseudomonas forment un large groupe colonisant le sol, les plantes et l'eau. Leur facilité de culture in vitro et la disponibilité d'un nombre croissant de séquences du génome de Pseudomonas ont fait de ce genre un foyer idéal pour la recherche scientifique. L'importance biologique fournie par les Pseudomonas dans le domaine de la recherche a donné naissance à un grand nombre d'informations. L'accumulation de ces informations dans des bases de données différentes a conduit à une hétérogénéité syntaxique et sémantique importante. Aujourd'hui, l'un des grands défis de la bioinformatique est de permettre aux biologistes d'accéder efficacement à plusieurs sources de données hétérogènes via des procédures automatiques. Dans ce cadre, notre travail a pour finalité la réalisation d'un environnement intégratif de données biologiques concernant les Pseudomonas. Ce travail entre dans le cadre d'une collaboration scientifique entre notre laboratoire de recherche LABIPHABE et le groupe KHAOS de l'université de Malage.

L'originalité de notre travail est de combiner l'approche matérialisée (entrepôt de données) et l'approche virtuelle (médiateur) pour profiter de ces avantages à la fois. L'entrepôt va permettre l'accès direct et rapide aux données alors que le médiateur permettra l'intégration de différentes sources de données et aussi il permettra la mise à jour des données en cas de besoin. Notre entrepôt de données nommé *PseudomonasDW* intègre les données biologiques stockées dans cinq bases de données différentes accessibles via le Web : Genbank, PRODORIC, UniProt, KEGG et BRENDA. *PseudomonasDW* est un entrepôt de données semi-structuré pour l'intégration sémantique des données du genre Pseudomonas. Il a été conçu dans le but de répondre aux besoins des biologistes en matière de données génomiques, protéomiques et métaboliques. L'intégration des données à partir des sources de données hétérogènes représente la consolidation des données hétérogènes conduisant à la reproduction des nouvelles données ne peuvent pas être obtenues à partir d'une seules source.

**Mot clés :** Pseudomonas, intégration de données, entrepôt, médiateur, approche hybride, PseudomonasDW.

# Remerciements

# Remerciements

Je tiens à adresser mes plus sincères remerciements au professeur Badr Din Rossi Hassani pour m'avoir accepté dans son laboratoire et intégré dans son équipe et de m'avoir encadré et aidé tout au long de ses années de thèse.

Je remercier également le professeur José F. Aldana Montes pour avoir accepté de Co-encadrer cette thèse, pour m'avoir accueilli si chaleureusement dans son équipe de recherche et pour m'avoir fait part de ses remarques pour mener à bien mes recherches.

Je remercie très sincèrement tous les membres du jury qui ont eu la lourde tâche de juger mon travail.

J'exprime toute ma profonde et sincère reconnaissance à tous les membres du groupe khaos. Je remercie tout particulièrement Ismael Navas Delgado; merci pour ton aide et ton précieux soutien.

A mon père et ma mère qui, malgré l'éloignement, ont cru en moi, m'ont toujours apporté leur soutien sans faille. Je les remercier de toute l'affection et tout l'amour qu'ils m'ont témoignés.

Toute ma reconnaissance et ma gratitude pour mon cher frère Mohamed qui m'a aidé avec une indéfectible patience. Merci pour ton amour inconditionnel et pour ton encouragement.

Merci à mon fiancé d'être toujours avec moi. Merci pour ton soutien régulier, tes compétences ainsi que ton intérêt pour la bioinformatique qui auront fortement contribué à l'avancement de ce travail.

Finalement, je tiens à remercier du fond du cœur, ma famille Marrakchi, mon petit frère Amine, ma belle–sœur Adiba qui a la position d'une vraie sœur ainsi que ses petits, ma grande mère « al haja », ma tante Doha, mon beau-père, ma belle-mère et toute la famille Briache.

Merci à tous ceux qui ont participé de près ou de loin à l'aboutissement de ce travail.

# Sommaire

# Sommaire

# INDEX DES FIGURES ET DES TABLES

## FIGURES

# TABLES

# ABREVIATION

# ABREVIATION

| | |
|---|---|
| ADN: | Acide Désoxyribonucléique |
| API: | Application Programming Interface |
| ASN: | Abstract Syntax Notation |
| BACIIS: | Biological And Chemical Information Integration System |
| BioGRID: | Biological General Repository for Interaction Datasets |
| BLAST: | Basic Local Alignment Search Tool |
| CGH: | Comparative genomic hybridization |
| ChEBI: | Chemical Entities of Biological Interest |
| CMR: | Comprehensive Microbial Resource |
| CPAN: | Réseau Complet d'Archives Perl |
| CPL: | Collection Programming Language |
| CSS: | Cascading Style Sheets |
| CSUQ: | Computer System Usability Questionnaire |
| CYGD: | Comprehensive Yeast Genome Database |
| DAML: | DARPA Agent Markup Language |
| dbEST: | Expressed Sequences Tags databases |
| DDBJ: | DNA Data Bank of Japan |
| DTD: | Document Type Definition |
| EBI: | European Bioinformatics Institute |
| EcoCyc: | Encyclopedia of Escherichia coli |
| EMBL: | European Molecular Biology Laboratory |
| EMBO: | European Molecular Biology Laboratory |
| EPG: | Entity Path Generator |
| ETL: | Extraction, transformation and loading |
| ExPASy: | (Expert Protein Analysis System |
| FTP: | File Transfer Protocol |
| GAM: | Generic Annotation Management |
| GAV: | Global As View |
| GDB: | Human Genome Databases |
| GEDAW: | Gene Expression DAta Warehouse |
| GenMapper: | Genetic Mapper |
| GEO: | Gene Expression Omnibus |
| GeWare: | Gene Expression Warehouse |

| | |
|---|---|
| GFF: | General Feature Format |
| GIMS: | Genome Information Management System |
| GLAV: | Generalized Local As View |
| GMOD: | Generic Modele Organisme Database project |
| GNU: | GNU's Not UNIX |
| GO: | Gene Ontology |
| GPL: | General Public License |
| GRAIL: | GALEN Representation and Integration Language |
| GUS: | Genomics Unified Schema |
| HGNC: | Human Gene Organisation |
| HGP: | Human Genome Project |
| HGP: | Human Genome Project |
| HTML: | HyperText Markup Language |
| HTTP: | Hypertext Transfer Protocol |
| IBM: | International Business Machines |
| ICARUS: | Interpreter of Commands And Recursive Syntax |
| IMG: | Integrated Microbial Genomes |
| INSDC: | Internatinal Nucleotide Sequence Database Collaboration |
| INSERM: | Institut National de la Santé et de la recherche médicale |
| IRISA: | Institut de Recherche en Informatique et Systèmes Aléatoires |
| JAXB: | Java Architecture for XML Binding |
| JAXP: | Java API for XML Processing |
| JDBC: | Java Database Connectivity |
| K2MDL: | K2 Mediator Definition Language |
| KEGG: | Kyoto Encyclopedia of Genes and Genomes |
| KOMF: | Khaos Ontology-based Mediation Framework |
| LAV: | Local As View |
| MCM: | Modèle Conceptuel Multidimensionnel |
| MeSH: | Medical Subject Headings |
| MGD: | Mouse Genome Database |
| MGI: | Mouse Genome Informatics |
| MIPS: | Munich Information Center for Protein Sequences |
| MOLAP: | Multidimensionnal On Line Analytical Processing |
| NAR: | Nucleic Acids Research |
| NBRF: | National Biomedical Research Foundation |
| NCBI: | National Center for Biotechnology Information |

| | |
|---|---|
| NIH: | National Institutes of Health |
| NXD: | Native XML Database |
| OBO: | Open Biomedical Ontologies |
| ODL: | Object Definition Language |
| ODMG: | Object Data Management Group |
| OIL: | Ontology Inference Layer |
| OLAP: | On Line Analytical Processing |
| OLTP: | On Line Transactionnel Processing |
| OMG: | Object Management Group |
| OMIM: | Online Mendelian Inheritance in Man |
| OOLAP: | Object On-Line Analytical Processing |
| OQL: | Object Query Language |
| OWL: | Web Ontology Language |
| PDP: | Protein Data Bank |
| Pfam: | Protein Famili |
| PHP: | Hypertext Preprocessor |
| PIR: | Protein Identification Ressource |
| PPI: | Protein-Protein Interaction |
| PQL: | Program Query Language |
| PRODORIC: | PROcariotIC Database Of Gene-Regulation |
| QUIS: | Questionnaire for User Interface Satisfaction |
| RDF: | Resource Description Framework |
| RDFS: | Resource Description Framework Schema |
| ROLAP: | Relational On-Line Analytical Processing |
| SB-KOM: | System Biology Khaos Ontology-based Mediator |
| SEPT: | Source Entity Path Translator |
| SGBD: | Système de gestion de base de données |
| SGD: | Saccharomyces Genome Database |
| SKB: | Source Knowledge Base |
| SOAP: | Simple Object Access Protocol |
| SOFG: | Standards and Ontologies for Functional Genomics |
| SQL: | Structured Query Language |
| SRS: | Sequence Retrival System |
| SUS: | System Usability Scale |
| Tambis: | Transparent Access to Multiple Bioinformatic InformationSources |
| TaO: | Tambis Ontology |

| | |
|---|---|
| UCL: | Université catholique de Louvain |
| UML: | Unified Modelling Language |
| UMLS: | Unified Medical Language System |
| UniProt: | Universal Protein Resource |
| URL: | Uniform Resource Locator |
| USA: | United States of America |
| W3C: | World Wide Web Consortium |
| WSDL: | Web Services Description Language |
| XML: | Extensible Markup Language |
| XSLT: | Extensible Stylesheet Language Transformations |
| ZFIN: | Zebrafish Information Network |

# NOTE AU LECTEUR

Dans la suite du document, les termes marqués par $\star$ seront définis dans le glossaire.

# INTRODUCTION GENERALE

## Intégration de données sur le Web : Etude générale et applications au domaine biologique

# Introduction générale

# Intégration de données sur le Web : Etude générale et applications au domaine biologique

Dès les premiers jours de l'ère de la génomique, la quantité de données a cru de manière exponentielle, conduisant à une émergence extraordinaire du nombre et du contenu des sources de données. L'ouverture de ces sources sur Internet* les a rendues disponibles au plus grand nombre, ouvrant ainsi de belles perspectives en recherche.

La diffusion des sources sur le Web*, s'est faite de manière indépendante, en séparant les données par entité biologique (ADN*, ARN*, Protéine*), par niveau d'organisation différent (cellules, tissus, organe, organisme, espèce*) et par technologie différente (analyse du transcriptome*, du protéome). Mais c'est la confrontation de toutes ces données diverses émanant de sources variées et jusqu'alors indépendantes qui va permettre de répondre à des questions biologiques complexes. L'effort consiste à intégrer des données hétérogènes afin d'en extraire de nouvelles connaissances, qui mènent à la découverte :

Données → Information → Connaissance → Découverte

La biologie prend ainsi une nouvelle dimension, anciennement divisée en plusieurs disciplines, elle devient intégrative et offre de belles perspectives d'appréhension de la complexité du monde vivant (Blagosklonny and Pardee, 2002).

Les phénomènes biologiques sont complexes et nécessitent la confrontation de différentes données. Ainsi, la compréhension des phénotypes normaux et pathologiques implique une prise en compte de données expérimentales, de données génomiques, de données issues des analyses bioinformatiques et de données de la littérature.

## 1 PROBLEMATIQUE ET MOTIVATION

Les pratiques concernant le stockage et la mise à disposition de données produites par les laboratoires de recherche ont évalué au cours du temps. Au début du stockage informatisé

des données, les résultats produits étaient sauvegardés localement, dans des bases de données développées et maintenues en interne, destinées uniquement à un usage personnel. L'accent était uniquement mis sur la sauvegarde rapide et fiable des résultats.

La prise en compte d'une ouverture future sur le monde (donc sur le Web) n'étant pas envisagée, les problématiques des accès et des modifications concurrentes, ainsi que la documentation destinée à l'utilisateur étaient souvent laissées de côté. En absence de consensus sur le modèle de donnée à utiliser, ou le langage de requêtes destiné à exploiter les enregistrements, les solutions individuelles se sont multipliées : formats binaires, fichiers plats, bases de données relationnelles, ou encore, bases de données objets et natives XML* (Harold and Means, 2004). Associés à ces bases de données, nous trouvons pêle-mêle les langages Perl* (Wall, 2000), SQL* (Lans, 1989), OQL* (Alashqur, et al., 1989), Xquery* (Katz, et al., 2003), ou simplement des adresses Web, qui à base de couples *clés-valeurs* sont parfois -trop souvent- le seul moyen d'extraire les informations qui intéressent le chercheur. Cette façon de procéder nous a amené à la situation que nous connaissons aujourd´hui avec des bases de données qui proposent certes souvent un format d'exportation commun (XML par exemple), mais dont les schémas sont hétérogènes, et les langages de requêtes incompatibles. La syntaxe et la sémantique* différent d'une base à l'autre, ce qui oblige l'utilisateur à un apprentissage préalable multiple : tant sur la signification des données enregistrées et des opérateurs que l'on peut leur appliquer, que sur la façon d'y accéder, par le biais de formulaires Web ou par une connexion directe au SGBD*.

De nos jours, la masse formidable de données produites par les centres de recherche atteint des quantités de plusieurs giga-octets par jour, entreposés dans une multitude de systèmes, répartis dans le monde entier ; à titre d'exemple, la version 176 de GenBank[1] (Feb 2010) occupe 463 giga-octets, et la version 188 (Feb 2012) occupe 580 giga-octets. Cette accumulation d'informations a engagé la biologie dans une phase de transition d'une science expérimentale à une science de plus en plus orientée par les données (Committee, 2005).

L'enregistrement des séquences* brutes, de la cartographie des chromosomes*, des données structurales ou dépression des gènes* ont obligé à apporter une attention toute particulière aux sources de données qui les contiennent. La connexion au Web ouvre ces sources à un nombre d'utilisateurs potentiellement illimité, même si en pratique, il est rare de dépasser le cap de plusieurs milliers de connexions simultanées. Cet état de fait oblige leurs concepteurs à une réflexion approfondie en amont, afin d'éviter l'asphyxie rapide du système, causée par la redondance, des structures de données inadaptées ou une mauvaise optimisation[2] qui font s'écrouler les performances lors d'un grand nombre d'accès. La

[1] http://www.ncbi.nlm.nih.gov/nuccore/

[2] La plupart des tables de la base *Ensembl* ont un index dont la taille dépasse celle des données elles-mêmes. La rapidité d'accès a été privilégiée - sciemment et avec succès - au détriment de l'espace de stockage Colonna, F.-M. (2008) Intégration de données hétérogènes et distribuées sur le Web et applications à la biologie. UNIVERSITÉ PAUL CÉZANNE AIX-MARSEILLE III..

majeure partie des sources basées sur des technologies éprouvées et robustes, comme des serveurs Oracle[3] (Ault, et al., 2003) ou MySQL[4] (Stephens and Russell, 2004) (souvent montrées en cluster*), donc aptes à répondre à une telle montée en charge.

L'un des principaux problèmes auxquels sont confrontés les biologistes aujourd'hui ne concerne donc plus la consultation individuelle d'une seule et unique source, mais plutôt l'interopération de plusieurs. Nous ne considérons dans la suite de cette introduction et la présentation de nos travaux que les sources de données qui correspondent aux critères décrits chaque année dans le journal Nucleic Acid research (Galperin and Fernández-Suárez, 2011), à savoir les banques de données ouvertes au public sans installation de logiciels complémentaires, et qui autorisent l'exploration de contenu stocké sans compensation financière[5].

Une des problématiques centrales des biologistes d'aujourd'hui consiste donc à rassembler les données extraites de plusieurs de ces sources, de façon la plus automatisée possible. Dans le cadre de nos travaux, nous nous sommes intéressés uniquement aux problèmes posés par l'intégration de données, que nous allons détailler un peu plus loin dans la suite de cette introduction. Un bon moyen de se rendre compte des difficultés éprouvées aujourd'hui pour la collecte de données consiste à s'intéresser à un scénario typique, résolu manuellement.

Considérons une question biologique simple à propos des réactions enzymatiques et les voies métaboliques auxquelles participe le produit d'un gène donné d'une espèce donnée :

« Quelles sont les réactions enzymatiques et les voies métaboliques auxquelles participe le produit du gène '*glpK1*' de l'espèce '*Pseudomonas aeruginosa PA7* ' ? »

Une réponse possible à cette question met en œuvre trois sources : la première étape consiste de chercher le nom du produit du gène par exemple dans la base de données Uniprot ( base de données protéique), et à reporter ensuite le nom de la protéine obtenu dans le formulaire de recherche proposé par la base de données de BRENDA[6] (par exemple) pour chercher les réactions enzymatiques et celui aussi de la base de données KEGG[7] pour chercher les voies métaboliques. Le croisement manuel des informations fournies individuellement nous apporte donc un ensemble de résultats, qui ne constitue qu'une partie des réponses possibles, puisque d'autres sources disponibles sur le Web nous auraient permis de répondre à cette même question. Le travail demander pour ce faible nombre de source est déjà fastidieux, et prend des proportions qui deviennent difficile à gérer à partir de cinq ou dix sources. Des simplifications existent, puisque des liens hypertexte permettent souvent de basculer d'une source à l'autre selon la valeur d'un

[3] http://www.oracle.com/index.html
[4] http://www.mysql.com/
[5] Des restrictions d'accès peuvent néanmoins exister afin de n'autoriser que certains types de requêtes.
[6] http://www.brenda-enzymes.info/
[7] http://www.genome.jp/kegg/

paramètre ; c'est notamment le cas dans les bases de données les plus connues telles que GenBank et Uniprot. D'un point de vue informatique, ces hyperliens entre objets hébergés dans des sources distribuées permettent d'obtenir une jointure, mais ces solution bien que très utiles pour collecter rapidement des données, sont insuffisantes : l'intervention humaine reste prépondérante ; de plus, l'expressivité de la requête est très limitée, pour ne pas dire inexistante.

Comme nous venons de l'évoquer, la diversité des formats, des interfaces, des langages de requêtes rend l'intégration de données (biologiques ou non) sur le Web difficile. Des solutions ont été proposées pour la collecte centrales de données au travers d'une interface unique : soit en exploitant les liens entre sources (intégration navigationnelle), soit dans le cadre des approches d'intégration matérialisées (entrepôt de données) ou virtuelles (architecture de médiation).

L'intégration navigationnelle consiste à regrouper les bases de données entre elles à partir des identifiants qu'elles partagent. Il s'agit de la méthode la plus simple, accessible à tous les utilisateurs sans apprentissage préalable. Elle reprend le principe appliqué lors de l'extraction manuelle, en sélectionnant les attributs à extraire de chacune des sources demandées.

Les deux dernières approches, la construction d'un entrepôt de données ou l'intégration de données virtuelle à l'aide de vues ont besoin toutes les deux d'un modèle de données* commun afin de représenter les données extraites des sources locales.

La démarche de création d'un entrepôt de donnés consiste à traduire massivement les données extraites des sources locales, afin de les rendre compatibles avec le modèle de données proposé à l'utilisateur. Cette adaptation des données présente un certain nombre d'inconvénients, tels que l'espace nécessaire au stockage et la mise à jour qui est très couteuse en temps et en trafic sur le réseau. Le système offre généralement un langage de requêtes qui permet d'appliquer des opérateurs d'extraction de données pour vérifier des hypothèses, ou bien réaliser des expérimentations *in silico*. Hammer et Schneider (Hammer J and Schneider M, 2003) vont jusqu'à préconiser la mise en place d'une seule et gigantesque base de données biologiques. Cette proposition s'apparente à de la science-fiction : l'espace physique occupé serait trop important, tant par les données que la conservation de leur traçabilité. Et les phases de mises à jour occuperaient la majorité du temps de fonctionnement du système.

La médiation de données permet d'intégrer uniquement les données souhaitées par l'utilisateur, qui exprime ses besoins au travers d'une requête posée sur un schéma global préalablement défini. Les données sont à jour en permanence, puisque relues à chaque fois qu'une nouvelle demande parvient au système. L'espace demandé pour stocker les données est faible, et dédié au mécanisme de mise en cache des requêtes s'il a été mis en place par les concepteurs. Les difficultés majeures de la médiation reposent essentiellement sur la

transformation de requêtes destinées aux sources de données locales, et la facilité d'évolution du schéma global en cas d'ajout ou de retrait d'une source, ce qui se produit très fréquemment sur le Web.

Les deux approches que nous venons d'évoquer se rejoignent par le fait que dans certains cas, les instances du schéma défini pour la médiation servent d'étape de transformation préalable au peuplement d'un entrepôt de données.

# 2 CADRE ET BUTS DU TRAVAIL

Les données biologiques reparties sur le Web sont nombreuses et de natures variées ; Il s'agit d'informations sur les séquences des gènes, leurs localisations chromosomiques, les protéines encodées, leurs distributions tissulaires, leurs implications dans des fonctions moléculaires et des processus biologiques, leurs implications cliniques, leurs niveaux d'expression dans différentes conditions physiopathologiques. Ajoutons à cela leur apparition croissante dans la littérature scientifique.

Un des défis actuels de la bioinformatique est de fournir des moyens pour intégrer cette masse de données et de l'exploiter de façon automatique pour en extraire de nouvelles connaissances. Cette tâche n'est pas triviale et révèle de nombreuses difficultés. En effet, comme démontré en partie introductive de ce manuscrit, ces données sont réparties sur le Web dans une multitude de sources de données dynamiques et très hétérogènes. Si depuis quelques années des efforts ont été fourni par la communauté scientifique pour améliorer l'interopérabilité* entre ces différentes sources par la définition de standards et la proposition de différentes approches d'intégration, la problématique reste entière.

Au cours de mon travail de thèse, mon objectif a été de fournir une solution d'intégration tenant compte des défis mentionnés ci-dessus et adaptée au contexte d'intégration de données biologique de l'espèce de *Pseudomonas*. L'enjeu était double :

- Intégrer des informations allant du gène à la pathologie et réconcilier ces données afin d'avoir une vue unifiée des informations disponibles sur une protéine donnée.
- Fournir une plateforme complète permettant d'orienter la recherche par extraction de nouvelles connaissances.

La première contribution de notre travail est l'utilisation d'une approche hybride (en combinant les avantages de l'approche virtuelle et ceux de l'approche matérialisée) pour la mise en place d'un système d'intégration semi-structuré appliqué dans le domaine biologique. Ce travail a été réalisé dans le cadre d'une collaboration scientifique entre notre

groupe de recherche *LABIPHABE* et le groupe de recherche *KHAOS* de l'université de Malaga.

La deuxième contribution de ce travail est la création d'un entrepôt de données biologique nommé '***PseudomonsDW'*** dédié aux espèces de *Pseudomonas*. L'un des volets d'intérêt de notre groupe de recherche *LABIPHABE* est l'étude de ce fameux micro-organisme. La section suivante décrit brièvement cette espèce. L'entrepôt de données ***PseudomonasDW*** intègre des données biologiques diverses (les gènes, les protéines, les enzymes, les sites de restrictions, les voies métaboliques…). Il est étendu par un Wiki scientifique nommé **PDWiki**. L'idée principale derrière **PDWiki** est de donner à la communauté scientifique de Pseudomonas de trouver, éditer et ajouter des informations relatives aux divers organismes, et aux différentes données intégrées dans ***PseudomonasDW***.

# 3 LES PSEUDOMONAS

## 3.1 Caracteres généreaux

Les bactéries du genre Pseudomonas sont des bacilles à Gram négatif (Eyquem, et al., 2005), mobiles par une ciliature polaire, rarement immobiles, non sporulés.

Ces bactéries chimio-organotrophes ont un métabolisme strictement respiratoire avec comme accepteur terminal d'électrons l'oxygène en aérobiose, et pour certaines espèces le nitrate en anaérobiose avec synthèse d'une nitrate-réductase (respiration de nitrate). Elles sont oxygène (+).

Les Pseudomonas sont caractérisés par la pluralité des substrats hydocarbonés utilisés comme source de carbone et d'énergie.

Ces bactéries sont très répandues dans la nature et caractérisées par leur résistance aux antibiotiques et aux antiseptiques.

### A) Morphologie et structure

Les Pseudomonas se présentent sous la forme de bâtonnets droits et fins 0,5 à 1,3 μm. La mobilité est très vive en aérobiose. La ciliature est polaire : monotriche – multitriche. Pour les espèces multitriches, le type de ciliature ne peut être établi que statistiquement en déterminant l'Indes flagellaire. Il peut varier selon les conditions de culture.

### B) Croissance et nutrition

De nombreuses espèces ou souches de Pseudomonas ne cultivent pas à 37°C, alors que la température de 30°C convient à tous, pathogènes et saprophytes.

La culture est facile sur milieu complexe avec ou sans production de pigment. Ils sont capables de cultiver sur des milieux minéraux synthétiques avec une source simple de carbone : acétale, pyruvate. Ces propriétés sont utilisées pour mettre en évidence les auxotrophies nécessaires pour l'identification (auxanogramme) par l'étude des substrats carbonés utilisables comme source d'énergie pour la croissance.

### C) Caractères physiologiques

Ces bactéries ont une longévité faible en culture, même à 4°C. Tous les modes de conservation possibles sont proposés : lyophilisation, eau distillée stérile avec une anse de culture à température ordinaire de 18°C (Pseudomonas phytopathogènes), gélose molle, tube à vis comme pour les Entérobactéries, congélation…

### D) Habita

C'est une bactérie ubiquiste qui vit normalement à l'état de saprophyte dans l'eau et le sol humide ou sur les végétaux. Elle résiste mal à la dessiccation. Cette bactérie peut survivre et se multiplier dans une infinie variété de liquides et de milieux, de supports et de matériels, surtout s'ils sont humides.

### E) Morphologie et caractères culturaux

Bacille à Gram négatif : 1 à 3 μm de long, 0,5 à 1 μm de large. Il est parfois entouré d'une pseudo-capsule appelée slime, qui peut jouer un rôle important dans la pathogénicité de cette bactérie.

Il peut être cultivé facilement sur tous les milieux en aérobiose (température de 37°C ou 30°C). Il dégage une odeur aromatique caractéristique de Pseudomonas seringa due à la production d'ortho-amino-acétophénone, intermédiaire du métabolisme du tryptophane et non liée à la production de pigment. Un milieu sélectif comme le milieu de Drigalski convient pour la culture.

### F) Aspects de colonies

Ils sont particuliers à cette espèce. Une dissociation spontanée en 3 types principaux peut être observée :

- Colonies ***LA*** (« large ») : isolées, grandes avec une partie centrale bombée et un contour irrégulier. Elles sont caractérisées par une autolyse qui donne un aspect métallique. Irisé lors de la culture en nappe de la bactérie. Ce phénomène est lié à l'action des enzymes protéolytiques bactériennes.
- Colonies ***SM*** (« small ») : petites, mates, légèrement bombées avec un bord circulaire régulier.

- Colonies ***M*** (muqueuse), bombées, opaques, visqueuses parfois coulantes. Ces colonies se rencontrent presque spécifiquement dans des infections chroniques, urinaires ou pulmonaires (mucoviscidose). La bactérie produit alors un polysaccharide extracellulaire (l'acide alginique) qui est différent du « slime ».

G) Production de pigments

C'est l'une des caractéristiques de cette espèce ; les pigments servent à son identification. Ils sont fluorescents ou non fluorescents.

**Pyoverdine**

Pigment jaune-vert fluorescent, soluble dans l'eau, insoluble dans le chloroforme, mis en évidence dans le milieu de King B (phosphate, sulfate, glycérol, peptone), sa production est inhibée par les ions sodium et favorisée dans les milieux carencés en fer.

Les Pseudomonas fluorescents se caractérisent par la production de composés fluorescents jaune-vert qui sont les sidérophores de ces bactéries. Les Pseudomonas aeruginosa produit en fait deux types de sidérophores : la pyochéline et 3 pyoverdines de nature chromopeptidique (Pa, PaA, PaB) de structure très voisine. Ces pyoverdines et, à un moindre degré, la pyochéline, sont excrétées par la bactérie et sont capable de chélater le fer et de le transporte.

**Pyocyanine**

Pigment bleu soluble dans l'eau et le chloroforme, caractéristique de P. aeruginosa qui est la seule espèce à le produire. La synthèse de ce pigment est diminuée en présence d'un excès d'ions phosphate et sodium. C'est un indicateur de pH, en solution à pH 3 = rouge, en milieu neutre ou alcalin = bleu. Il peut jouer le rôle d'accepteur terminal d'électrons si la chaîne respiratoire est inhibée par exemple par l'azide de Na.

Il existe des souches de P. aeruginosa apigmentées : moins de 5% des souches sauvages ne produisent aucun de ces pigments. Elles sont fréquemment isolées chez des malades traités aux antibiotiques.

Il faut noter que d'autre Pseudomonas et apparentés produisent des pigments souvent de couleur jaune, notamment des espèces phytopathogènes, et il convient d'en faire le diagnostic différentiel : p. fluorescens, P. putida, P. aureofaciens, P. chlororaphis, P. lemonieri, P. stutzeri et P. mendocina.

## 3.2 Pouvoir pathogène

**Chez l'homme**, l'espèce Pseudomonas aeruginosa intervient fréquemment comme pathogène opportuniste. Elle se retrouve en flore de transit sur la peau et les muqueuses et

cause des surinfections de plaies ou brûlures. Chez des individus immunodépressifs elle peut être la cause de diverses infections cutanées et viscérales voire de septicémie. Elle comporte un risque particulièrement élevé d'infections nosocomiales (contractées par l'intermédiaire de soins en milieu hospitalier), notamment avec des souches résistantes à certains antibiotiques courants.

**Chez les plantes,** Pseudomonas syringae est un pathogène prolifique. Elle semble « opportuniste ». Elle infecte des plantes déjà affaiblie par la pollution, un stress hydrique, de mauvaises conditions de plantation, une autre maladie, des blessures, un système racinaire contraint ou asphyxié.

Il existe de nombreuses autres espèces de Pseudomonas qui peuvent agir comme agents pathogènes des plantes, notamment tous les autres membres du sous-groupe de Pseudomonas syringae, mais Pseudomonas syringae est la plus répandue et la mieux étudiée.

### 3.3 Lutte biologique

De nombreuses souches de Pseudomonas jouent un rôle majeur dans les processus de biodégradation. Dans les processus de remédiation et traitement de sites pollués, la biodégradation ou peut être favorisée ou accélérée par des apports en nutriments ou par des souches bactériennes sélectionnnées. C'est le cas par exemple pour les pollutions du sol ou de l'eau par du fuel ou du pétrole brut. Dans ce cas un ensemencement par des souches mixtes de Pseudomonas et de Rhodococcus et se sont montrées plus efficaces pour dégrader le fuel en milieu aquatique. Dans ce dernier cas, on n'a pas réussi a améliorer les performances des bactéries en portant l'association à trois, quatre, ou cinq souches d'autres bactéries.

**Dans le sol**, les Pseudomonas représentent une grande fraction de la communauté microbienne partageant leur milieu avec des commensaux représentant principalement les genres Bacillus et Actinomyces. On les retrouve sous tous les horizons, particulièrement sur les systèmes racinaires des plantes. Les différentes espèces de Pseudomonas qui colonisent la rhizosphère possèdent plusieurs caractéristiques intrinsèques qui les rendent particulièrement intéressantes pour une utilisation comme agents de lutte biologique. Premièrement, leur capacité à coloniser les racines et à y maintenir une forte densité de population est remarquable (Haas and Keel, 2003). Cette grande rhizocompétence vient sans doute de leur taux de croissance plus élevé que celui de la plupart des autres rhizobactéries et de leur capacité à métaboliser efficacement plusieurs composants des exsudats racinaires (Chin-A-Woeng, et al., 2000). De plus, ces bactéries sont très faciles à isoler et à cultiver au laboratoire et se prêtent aisément aux manipulations génétiques (Chin-A-Woeng, et al., 2001).

Les Pseudomonas, principalement l'espèce Pseudomonas fluorescens, sont connues depuis longtemps pour leur aptitude à réduire l'incidence des maladies racinaires dans certains champs, ainsi qu'à inhiber la croissance d'un grand nombre d'agents phytopathogènes in vitro. Cette capacité d'inhibition peut se faire selon plusieurs mécanismes incluant la production d'une large gamme de métabolites antagonistes et de sidérophores. Ces derniers permettent de compétitionner farouchement pour l'acquisition du fer. Dans un milieu comme le sol où cet élément est présent en très faible quantité, cela peut nuire à la croissance de plusieurs agents pathogènes et ainsi réduire la sévérité de la maladie.

# 4 STRUCTURE DE DOCUMENT

Dans le premier chapitre de cette thèse, nous présentons et nous mettons en évidence les différentes caractéristiques des sources de données biologiques. Ce chapitre comporte une description des divers niveaux d'hétérogénéité entre les sources.

Le deuxième chapitre dresse un état de l'art qui illustre chacune des solutions majoritairement suivies en informatique (entrepôt, médiateur et système navigationnel) et montre comment elles ont été appliquées aux données biologiques.

Le chapitre trois introduise notre solution hybride et présente les différentes étapes de la mise en place d'un nouveau système d'intégration concernant les données biologiques des espèces de Pseudomonas. Ce chapitre décrive l'outil ETL (Thomas and Stefan, 2008) qui permet l'extraction, la transformation et le stockage de données à partir des sources de données originales jusqu'à ***PseudomonasDW***.

Le chapitre quatre de cette thèse présente une nouvelle base de données pour les espèces de Pseudomonas. Ce chapitre comporte, en outre, une section qui décrive les phases de l'implémentation de notre base de données et l'interface utilisateur qui permet aux utilisateurs d'accéder aux données de ***PseudomonasDW***. Dans ce chapitre, nous détaillons aussi le processus d'intégration de quelques outils bioinformatique dans ***PseudomonasDW*** et de développement du wiki scientifique qui permit à l'utilisateur d'éditer, d'ajouter et d'annoter les données intégrées dans ***PseudomonasDW***.

Enfin, nous concluons le travail en ouvrant des perspectives sur nos travaux de futurs.

# Première Partie

# CHAPITRE 1

# Hétérogénéité et intégration de données : état de l'art

# Chapitre 1

# Hétérogénéité et intégration de données : état de l'art

**Sommaire**



## 1 INTRODUCTION

Ce chapitre est dédié à la présentation des sources de données biologiques. Notre objectif est de mettre en évidence les particularités de ces sources et de motiver le besoin de solutions d'intégration adaptées à ces types de données.

Les premières sources de séquences biologiques sont apparues dans les années 80 sous l'initiative de quelques équipes comme celle du Professeur Grantham à Lyon (Gautier, 1981). Avec les évolutions techniques du séquençage*, la gestion des données a nécessité une organisation plus conséquente. Ainsi, plusieurs organismes ont pris en charge la mise en place de systèmes de stockage des données.

En Europe, une équipe financée par l'EMBO[8] a développé une source de séquences nucléiques, l'EMBL data library (Hamm and Cameron, 1986). Du côté américain, soutenue par le NIH[9], la source nucléique *GenBank* a été créée à Los Alamos (Bilofsky and Christian, 1988). Cette source était à l'origine une base de données relationnelle puis fut diffusée sous la forme de fichiers plats par le NCBI[10]. La collaboration entre les concepteurs d'EMBL et de GenBank a commencé relativement tôt. Elle s'est étendue en 1987 avec la participation de la DDBJ[11] (Dna Data Bank) du Japon pour proposer en 1990 un format unique de description des caractéristiques biologiques qui accompagnent les séquences dans les sources de données nucléiques.

Pour les protéines, deux sources principales ont rapidement été créées. La première, sous l'influence du NBRF à Washington, est PIR, Protein Identification Ressource (Sidman, et al., 1988). La deuxième, SwissProt, a été développée à l'Université de Genève dès 1986.

# 2 ÉTAT DES SOURCES

Durant ces 20 dernières années, les sources de données biologiques disponibles sur le Web étaient multipliées. Leur croissance est en très forte progression depuis 10 ans. La '*Databases Issue*' de la revue Nucleic Acids Research (NAR), qui liste chaque année les sources les plus importantes du Web, recense plus de 1380 sources publiques en 2012 (Galperin and Fernández-Suárez, 2012). Ces sources étaient environ 1330 en 2011 et un peu moins de 1230 en 2010. En l'espace de 2 ans plus de 150 sources de données publiques ont donc vu le jour.

On peut proposer trois éléments d'explication à ce phénomène. D'abord, depuis les dix dernières années, les projets de séquençage étaient extrêmement développés. Chacun de ces projets a pour but de séquencer un génome*; il conçoit et développe alors sa propre source de données pour mettre ses résultats à la disposition de tout le monde. Citons le Human Genome Project (HGP) débuté en 1990 et le Mouse Genome Database (MGD) quelques années plus tard comme exemples de projets d'annotation* ayant mis en ligne leurs résultats. En parallèle, de nouvelles techniques d'analyse biologique à haut débit ont vu le jour, comme les puces à ADN et plus récemment les puces à protéines* ou les puces à CGH*. Ces nouvelles techniques ont généré de nouveaux types de données, qui ont été stockés dans de nouvelles sources. Ainsi, les sources GEO[12] et ArrayExpress[13] ont été

[8] http://www.embo.org/
[9] http://www.nih.gov/
[10] http://www.ncbi.nlm.nih.gov/
[11] http://www.ddbj.nig.ac.jp/
[12] http://www.ncbi.nlm.nih.gov/geo/
[13] http://www.ebi.ac.uk/arrayexpress/

créées pour contenir des données de puces à ADN (microarray). La troisième cause est le développement d'outils bioinformatiques. Les données sont aujourd'hui régulièrement analysées et comparées à l'aide d'outils de recherche de similarités de séquence (Blast[14]*), d'alignements multiples, ou encore de détection de gènes dans les séquences...etc. Les résultats obtenus par ces outils sont eux aussi stockés dans de nouvelles sources de données. Par exemple, la source Pfam[15] contient des données-résultats d'alignements multiples.

La sous-section suivante dresse un rapide panorama d'un certain nombre de sources de données que l'on peut trouver aujourd'hui sur le Web.

## 2.1 Variété des sources biologiques

Il n'existe à l'heure actuelle aucune classification suivie des sources de données. La classification proposée dans la revue NAR n'est, par exemple, pas la même d'une année à l'autre (les catégories changent) et regroupe les sources en fonction du type de données qu'elles contiennent (séquences...) ou de l'espèce concernée. À travers la (très simple) classification ci-dessous, nous ne cherchons pas être exhaustifs ni à proposer des classes (de sources) disjointes mais simplement à donner un aperçu des familles de sources de données biologiques publiques. Nous nous sommes inspirés de la revue NAR et des travaux de Carole Goble (Goble, 2002). Nous considérerons donc les familles de sources suivantes :

- Les sources regroupant un ensemble d'abstracts de **publications scientifiques** du domaine médical : Medline[16], PubMed[17].

- **Les sources de données primaires**. Ces sources sont les plus volumineuses. Il en existe essentiellement pour deux types de données à l'heure actuelle : (i) les séquences génomiques et (ii) les données de puces à ADN. Les sources GenBank (USA), EMBL (Europe), et DDBJ (Japon) sont des dépôts de séquences, qui contiennent toutes les trois les mêmes données, et sont mises à jour toutes les nuits les unes par rapport aux autres. Pour les données de puces à ADN, les dépôts de données sont ArrayExpress (Europe) et GEO (USA).

  Le rôle d'un dépôt est de contenir de façon exhaustive l'ensemble des données disponibles (sur les séquences ou les données de puce à ADN*). Plus précisément, chaque nouvelle séquence (ou nouvelle expérience de puce à ADN) découverte par

[14] http://blast.ncbi.nlm.nih.gov/Blast.cgi
[15] http://pfam.sanger.ac.uk/
[16] http://www.medline.com/
[17] http://www.ncbi.nlm.nih.gov/pubmed/

un laboratoire doit être envoyée à GenBank/EMBL/DDBJ (ou GEO/ArrayExpress) dans un certain format. Toute publication scientifique soumise à une revue en biologie au sujet d'un séquençage (ou d'une expérience de puce à ADN) doit être associée à un ou plusieurs numéros d'identification GenBank/EMBL/DDBJ (respectivement GEO/ArrayExpress).

Les données qui sont présentes dans ces bases, sont donc brutes au sens où elles ne sont pas validées par les propriétaires des sources. Il arrive même que des séquences soient dupliquées par erreur de manipulation des chercheurs lors de la soumission.

- **Les sources de données secondaires**. Contrairement aux précédentes, ces sources contiennent des informations nettoyées (au moins automatiquement, comme la suppression de doublons...) et parfois même validées manuellement par des experts. Ces sources sont dites secondaires car l'objectif de leurs propriétaires est de partir de données issues des sources primaires pour proposer des informations plus synthétiques et le cas échéant, ajouter des informations complémentaires.

  Pour les données génomiques, les sources RefSeq[18] et UniGene[19] du NCBI[20] sont deux exemples de sources secondaires, qui proposent de regrouper les fiches GenBank. La première propose une version non redondante de GenBank, elle est obtenue en utilisant des techniques de regroupement semi-automatiques, alors que la seconde construit de façon automatique des clusters de séquences.

- **Les sources de données d'expertises**. Ces sources contiennent essentiellement du texte et proposent des fichiers contenant une analyse et une synthèse d'un ensemble d'articles scientifiques. Par exemple, la source OMIM[21] fournit un ensemble d'informations sur les maladies humaines sous la forme de fichiers dans lesquelles des experts (de l'université Johns Hopkins aux USA) commentent les résultats associés à un gène ou un groupe de gènes décrits dans un ensemble de publications, et associés à un phénotype* (une maladie) donné.

- **Les sources de données-résultats d'outils**. On retrouve beaucoup de ces sources au niveau du recensement des domaines fonctionnels : Pfam, ProDom[22], Genopage (Cohen-Boulakia, et al., 2002). Ces sources ont des contenus générés automatiquement qui résultent de l'utilisation d'une succession précise d'outils bioinformatiques. Elles sont ensuite validées ou non par des experts. Ces sources

[18] http://www.ncbi.nlm.nih.gov/RefSeq/
[19] http://www.ncbi.nlm.nih.gov/unigene
[20] http://www.ncbi.nlm.nih.gov/
[21] http://www.omim.org/
[22] http://prodom.prabi.fr/prodom/current/html/home.php

sont aussi caractérisées par le fait qu'elles offrent des outils de visualisation des résultats, qui permettent de comparer et d'analyser les informations ainsi générées.

- Les sources qui offrent un degré élevé de précision sur **une famille de données** :
  - sur une famille de fonctions biologiques. Par exemple, la source BRENDA est dédiée à la description des protéines dont la fonction est enzymatique.
  - sur une espèce particulière, ou une famille d'espèces comme les sources FlyBase[23] (dédiée à la drosophile) et Saccharomyces Genome Database, SGD[24] (dédiée à la levure).

- Enfin, on distinguera les sources **synthétiques** qui proposent un ensemble de fichiers de synthèse. Chacune de ces fichiers regroupe des informations présentes dans d'autres sources associées à un même gène ou une même protéine. On trouve dans cette catégorie GeneCards[25] (Rebhan, et al., 1997), qui fournit des fichiers de synthèse proposant des liens hypertextes vers des informations relatives aux gènes humains, qui proviennent d'une vingtaine de sources de données (dont UniProt (Consortium, 2010), GenBank...).

## 2.2 Autonomie et capacités d'interrogation

La majorité des sources disponibles sur internet fonctionnent en mode totalement autonome. Autrement dit, les administrateurs et curateurs de ces sources sont tout à fait libres de modifier leur schéma ou de mettre à jour leur contenu (ces sources fonctionnent souvent sur le principe de mises à jour régulières, comme UniProt par exemple) sans en faire état préalablement aux utilisateurs. Aucune source ne tient compte des éventuelles références dont elle est l'objet ; or, en intégration de données, l'indisponibilité d'une source pendant sa maintenance va influer plus ou moins fortement sur la qualité et la complétude du résultat d'une requête, problème qu'un outil d'intégration de données du Web doit prendre en compte et résoudre, ou tout au moins signaler à l'utilisateur. La seule solution afin d'avoir en permanence les données intégrées les plus à jour, est d'accéder à celles-ci lors de l'exécution des requêtes.

Un facteur d'inconsistance supplémentaire des sources de données orientées Web est leur grande dépendance vis-à-vis du réseau. Les performances des transferts sur internet étant imprévisibles, "*n'importe quel système d'intégration qui accède à des données du Web hérite de cette imprévision*" comme l'ont souligné Jagadish et Olken (Jagadish and Olken, 2003). Les accès aux données peuvent être effectués via un navigateur HTTP ou un logiciel client

[23] http://flybase.org/
[24] http://www.yeastgenome.org/
[25] http://www.genecards.org/

FTP, par connexion directe sur la base de données (client dédié ou JDBC (Reese, 2001) par exemple), ou plus récemment encore via des appels de services Web. Concernant les interfaces homme-machine, chaque source propose ses propres fonctionnalités, ce qui suppose et impose à l'utilisateur une phase d'apprentissage pour chacune des interfaces qu'il devra utiliser.

Des restrictions d'accès existent sur les sources, et certaines requêtes ne peuvent, tout simplement, pas être exécutées. Ces limitations empêchent dans certains cas l'extraction d'informations pertinentes, même si les données pour y répondre sont disponibles (Sujansky, 2001). Les motivations de ces choix s'expliquent :

- soit par la volonté d'assurer une qualité de service identique à tous les utilisateurs : il n'est donc pas envisageable qu'un seul d'entre eux mobilise des heures durant la puissance de calcul d'une source par une requête trop complexe

- soit pour des raisons de droits de copie des données : l'extraction massive d'informations est alors limitée volontairement par les propriétaires de la source

Souvent, les langages de requêtes proposés n'en sont pas réellement : le système d'interrogation est constitué uniquement d'un index de taille plus ou moins importante, et via des formulaires accessibles dans des pages HTML*, va chercher dans une ou plusieurs sources les valeurs associées aux attributs choisis. Des langages de plus haut niveau plus expressifs sont également utilisés, tels que SQL ou OQL.

L'intégration ne doit d'ailleurs pas simplement concerner les données brutes, mais aussi permettre l'utilisation de ressources biologiques, telles que Blast(Altschul, et al., 1990), ou Fasta*[26] (Lipman and Pearson, 1985).

L'autonomie des sources les unes par rapport aux autres, l'hétérogénéité de leurs représentations, mais aussi les interfaces d'accès différentes et aux capacités d'interrogation inégales rendent difficile, voire impossible leur utilisation combinée par des biologistes. Les procédures permettant de collecter les données doivent autant que possible être automatisées, et c'est cette tâche qui échoit au système d'intégration, avec plus ou moins de facilité en fonction de l'approche suivie.

[26] http://www.ebi.ac.uk/Tools/sss/fasta/

# 3 DIFFICULTES RENCONTREES LORS DE L'INTERROGATION DES SOURCES

Le nombre de sources de données et d'outils mis à la disposition des biologistes sur le Web n'a cessé de croître ces dernières années. Cette augmentation colossale de la masse de données disponibles a généré une grande variété d'interfaces d'accès, mais aussi et surtout une profonde hétérogénéité syntaxique et sémantique. Jusqu'à présent, les recoupements effectués par les biologistes entre plusieurs sources de données étaient réalisés à la main, au cas par cas. Les interrogations des sources devaient se faire une à une, puis dans l'ensemble de résultats obtenus, il fallait faire la part des redondances et des complémentarités, ainsi que des éventuelles inconsistances. Désormais, la compréhension des processus globaux des phénomènes vitaux doit faire appel à une automatisation des traitements.

En évoluant indépendamment, les sources ont adopté chacune leur propre modèle de données, leur langage de requêtes, et leur format d'exportation, que la littérature a détaillé à de nombreuses reprises (Davidson, et al., 1995; Hernandez and Kambhampati, 2004; Olken and Jagadish, 2003). La résolution de ces conflits est l'objectif de nombreuses approches qui diffèrent par les méthodes et les moyens qu'elles mettent en œuvre. La taxonomie* des conflits peut être définie suivant quatre grandes dimensions de variation, mais celles-ci ne sont pas spécifiques et limitées au domaine biologique, puisque des problématiques similaires se retrouvent également en géographie par exemple (Aerts, et al., 2006; Bishr, 1998). Nous allons énumérer ici les quatre propriétés des sources biologiques qui rendent leur interrogation complexe et fastidieuse.

## 3.1 Diversité syntaxique

L'hétérogénéité syntaxique est causée par les différences entre plateformes logicielles, et les formats qu'elles manipulent. Des informations identiques peuvent donc être enregistrées soit en utilisant des notations formelles telles qu'ASN 1.0[27] ou Fasta (Lipman and Pearson, 1985), soit du XML, du HTML ou des SGBD relationnels ou objets.

L'utilisation de fichiers plats est le standard *de facto*, ce qui nécessite une phase d'extraction de données afin de retrouver la structure des données originelles. Le développement du langage XML et des technologies qui y sont liées (notamment autour du langage Java avec par exemple les API* JAXP (Griffith, 2005) et JAXB (McLaughlin, 2002)) permet de plus en plus de simplifier les échanges de données biologiques (Achard, et al., 2001). L'interprétation de l'information intégrée reste malgré tout un problème crucial à résoudre.

[27] http://www.bgbm.org/tdwg/acc/Documents/asn1gloss.htm

### 3.2 Diversité sémantique

- Diversité des schémas : Dans cette partie, nous allons exposer des problèmes qui sont plus propres aux données biologiques que ceux listés ci-dessus.
    - Diversité des focus : Chaque source se focalise sur un type d'objet, une entité biologique. Dans UniProt, les données sont focalisées sur la protéine, qui est l'entité centrale : toute entré de UniProt décrit une protéine. Le gène codant pour chaque protéine est alors vu comme un simple attribut. Au contraire, dans GenBank, la séquence nucléotidique est l'entité centrale et c'est la protéine qui en est un attribut. L'entité centrale peut aussi être le domaine fonctionnel (dans InterPro[28]) ou la structure 3D d'une protéine (dans PDB[29]).
    - Diversité du niveau de granularité : selon les sources, une même donnée n'est pas représentée avec le même niveau de granularité, de détail. Par exemple, UniProt propose des informations sur des protéines issues de différentes espèces. Elles sont précises mais "généralistes" au sens où elles ne sont pas ciblées sur une famille particulière de données. Au contraire, chez SGD on pourra connaître de façon spécifique la fonction de chacune des protéines de la levure.
    - Diversité dans la définition biologique d'une entité. Selon les sources, une même entité biologique (gène, protéine ...) est définie différemment. Par exemple, selon les sources, une protéine est une isoforme particulière (GenBank) ou bien la séquence associée à l'ensemble des isoformes (UniProt). On a le même problème au niveau de la définition d'un gène qui peut varier : considération de la séquence codante (après épissage) ou incluant les introns*.

*La diversité des sources de données permet au biologiste d'accéder à des informations complémentaires mais qui peuvent être très redondantes : selon la source, une même information peut être représentée avec des modèles, des formats et des schémas différents.*

- Diversité des informations au niveau des instances :
    - Différents points de vue sur les données. Chaque annotateur exprime son expertise à travers une fiche. Il peut arriver que, selon les sources, une même protéine soit associée à des fonctions différentes.
    - Différents vocabulaires pour annoter les séquences. Le degré de confiance associé aux annotations n'est pas souvent donné dans les sources, et il est peu homogène au sein même d'une source voire à l'intérieur d'une équipe d'annotateurs. Certains annotateurs emploieront le terme de "putative"

[28] http://www.ebi.ac.uk/interpro/
[29] http://www.rcsb.org/pdb/home/home.do

pour exprimer que l'annotation n'est pas sûre, tandis que d'autres utiliseront le terme "hypothetical". D'autres encore ne préciseront rien.

- Différents noms pour un gène ou une protéine : il existe très souvent plusieurs noms (synonymes) pour un même gène ou pour une même protéine, et ce, à l'intérieur d'une même source mais aussi à travers les sources et les espèces. Il est donc courant qu'un gène ou une protéine ait plusieurs noms. De même, il est possible que deux protéines ou deux gènes différents aient le même nom ou un nom en commun : on est dans ce cas en présence d'homonymie.

*L'information présente dans les sources au niveau des instances est donc complémentaire mais elle peut aussi être divergente. Les homonymies peuvent conduire à de fausses divergences alors que les différents points de vue d'experts peuvent refléter de réels désaccords. Face à des informations divergentes, le biologiste privilégie les informations issues de la source en laquelle il a le plus confiance (notons que cette confiance est variable, puisqu'elle peut dépendre du domaine de recherche voire de l'expérience qu'a un biologiste de l'utilisation de la source). Il est donc primordial que le biologiste sache de quelles sources proviennent les données.*

### 3.3 Diversité des langages de requête

Il découle de la sous-section 3.1 que les sources ont des langages de requêtes différents. Le langage d'interrogation d'une banque de données (comme PubMed/Medline, GenBank...) est souvent une simple combinaison de mots à chercher dans les textes tandis que les bases de données relationnelles par exemple, peuvent être interrogées en SQL (c'est le cas pour la source ensEMBL[30]). Certains projets d'entrepôts orientés-objet (comme GEDAW (Guérin, et al., 2005) ou GIMS (Cornell, et al., 2003)) offrent la possibilité de poser des requêtes OQL sur leur schéma.

### 3.4 Diversité des services

Les sources proposent des outils capables de rechercher certaines propriétés des données (le plus souvent, ces outils servent à renvoyer les données d'une source qui sont similaires à une donnée expérimentale présentée en entrée). Une forte diversité est présente à travers ces outils : chaque source, possède une ou plusieurs variantes d'un même outil ; en outre, l'utilisateur dispose très rarement d'une description complète de l'outil qu'il manipule. Par exemple, dans le cas d'un Blast, il existe des variantes de l'algorithme considérant des heuristiques différentes, ou tout simplement des algorithmes adaptés à des types de

[30] http://www.ensembl.org/index.html

données différents (séquences d'acides aminés* comme BlastP ou de séquences nucléotidiques comme BlastN).

# 4 ELEMENTS DE STANDARDISATION

Dans la mise en place d'éléments de standardisation, trois types de solutions ont été proposés. Le premier est relatif à la modélisation du contenu des sources : **choix des noms des concepts sous-jacents aux données des sources et des noms des relations entre ces concepts**. Cette tâche ne peut se faire qu'à travers de nombreuses discussions entre experts; ce type de solution est donc spécifique à chaque domaine de connaissance. Le second type de solution est plus générique, il comprend **la construction de cadres de représentation et d'échange des concepts et de leurs relations ainsi que l'élaboration de méthodes pour faire correspondre des ensembles structurés de concepts développés dans des contextes différents**. Enfin, un troisième type de solutions a été proposé, il vise à **ajouter des informations à propos des données contenues dans les sources**, on parle alors du développement de métadonnées.

## 4.1 Format standards et nomenclatures

Un premier élément de solution pour l'intégration des données est l'établissement de terminologies standards pour décrire les données.

Dans le domaine biologique, plusieurs consortiums se sont formés en vue d'établir des terminologies pour décrire les données présentes dans les sources et des hiérarchies pour classifier les concepts sous-jacents à ces terminologies. Depuis quelques années un workshop "Standards and Ontologies for Functional Genomics" (SOFG) a lieu annuellement et regroupe les principaux acteurs sur cette problématique.

Le souci de standardisation de l'attribution de noms est pris en compte par le consortium HGNC[31] (Human gene organisation (HUGO) Gene Nomenclature Committee) qui propose une terminologie particulière pour les nouvelles séquences.

[31] http://www.genenames.org/

## 4.2 Ontologies

Le besoin de capturer les notions biologiques présentes à travers le Web et de traiter de façon automatique des annotations généralement écrites en langage naturel a conduit à la construction de nombreuses ontologies.

Le concept d'ontologie est employé dans des domaines très différents tels que la philosophie, la linguistique ou l'intelligence artificielle. L'une des premières définitions informatiques de cette notion comme celle de Gruber (Gruber, 1995) est "*spécification d'une conceptualisation*". Outre le sens philosophique originel, une ontologie désigne donc le plus souvent un ensemble structuré de concepts. À la différence d'un *vocabulaire*, une ontologie cherche à représenter le sens des concepts et des relations qui les lient. Une ontologie a donc deux composantes : (i) un ensemble de concepts et (ii) un langage pour structurer ces concepts.

Nous donnons ci-dessous un aperçu des ontologies développées dans le domaine biologique :

Tout d'abord, citons le projet GO[32] (Gene Ontology) (Ashburner, et al., 2000) qui vise à fournir un ensemble structuré de vocabulaires pour des domaines biologiques spécifiques permettant de décrire des produits de gènes (protéines ou ARNs) dans un organisme eucaryote* donné. GO est composée de trois ontologies respectivement consacrées aux fonctions moléculaires, aux processus biologiques et aux composants cellulaires. Il est à noter que GO est aujourd'hui très couramment utilisée par la communauté des biologistes qui travaillent sur des organismes eucaryotes. D'autres ontologies, plus spécifiques, sont utilisées pour les procaryotes. C'est le cas de l'ontologie MIPS (Mewes, et al., 2002) ou l'ontologie SubtiLis (Moszer, et al., 2002).

Beaucoup d'autres ontologies ont été développées ; le projet OBO[33] (Open Biomedical Ontologies) (Xuan, et al., 2009) liste notamment l'ensemble des ontologies en ligne dont voici un extrait.

- Pour modéliser des organismes, des ontologies sur l'anatomie d'espèces particulières ont été proposées comme MGI[34] (Mouse Genome Informatics) du *Jackson Laboratory*, Flybase du *Flybase Consortium*, ou encore ZFIN[35] (Zebrafish Information Network) du *groupe Zebrafish*. Dans la communauté biomédicale, on distinguera l'UMLS[36] (Unified Medical Language System), un méta-thesaurus pour

[32] http://www.geneontology.org/
[33] http://www.obofoundry.org/
[34] http://www.informatics.jax.org/
[35] http://zfin.org/
[36] http://www.nlm.nih.gov/research/umls/

les concepts manipulés en médecine ou encore le MeSH[37] (Medical Subject Headings) qui contient essentiellement des termes pour l'anatomie humaine.

- Au niveau des voies métaboliques, la source de données KEGG (Kanehisa, et al., 2004) a développé sa propre ontologie. On trouve aussi EcoCyc[38] et MetaCyc[39] (Karp, et al., 2000) de P. Karp et ChEBI[40] (Brooksbank, et al., 2005), un dictionnaire pour les entités chimiques et une ontologie associée, développés à l'EBI[41].

- Pour représenter les structures des composants du ribosome, RiboWeb[42] (Chen, et al., 1997) propose un format de données, une nomenclature et un cadre XML (RNA-ML) (Waugh, et al., 2002).

Néanmoins, ces ontologies, même dans un domaine fixé (par exemple en anatomie) n'ont pas les mêmes structures de données sous-jacentes. Ainsi, les anatomies dans ZFIN et MGI sont représentées par une structure d'arbres alors que dans FlyBase, les ontologies se présentent sous la forme de graphes non cycliques.

### 4.3 Métadonnées

Il n'existe pas de définition consensuelle sur ce qu'est une métadonnée hormis le fait qu'il s'agit d'une information de niveau supérieur sur des données ou de toute donnée associée à une ressource permettant de décrire, sous divers aspects, cette ressource. Une métadonnée permet de donner du sens au contenu des ressources de manière à ce que leurs localisation et interrogation soient plus aisées et plus pertinentes. On peut citer de nombreux exemples de métadonnées :

- l'auteur de la ressource, sa date de création, sa date de dernière modification,
- des commentaires exprimant un point de vue sur la ressource,
- le schéma des données, les index associés,
- des informations de qualité relatives au schéma de la ressource,
- des informations statistiques sur les données,
- la spécification, la signature d'un programme,...

[37] http://www.nlm.nih.gov/mesh/
[38] http://ecocyc.org/
[39] http://metacyc.org/
[40] http://www.ebi.ac.uk/chebi/
[41] http://www.ebi.ac.uk/
[42] http://helix-web.stanford.edu/riboweb.html

Pour structurer et donner un sens aux métadonnées, plusieurs normes ont été proposées. Malgré leurs différences, leur objectif est d'uniformiser la manière d'effectuer la description des ressources et donc d'améliorer leur échange et leur partage. De manière générale, les normes proposent un guide de structuration des métadonnées nécessaires à la description d'une ressource. Les métadonnées sont présentées sous forme d'éléments (sections ou rubriques), lesquels peuvent, selon leur sémantique, être regroupés en catégories.

Par exemple, la norme Dublin Core[43] propose 15 éléments de description (métadonnées) d'une ressource organisés en trois catégories concernant :

- **le contenu de la ressource** : titre, sujet ou codes de classement, description, source, langue, relation avec une autre ressource, couverture spatiale et temporelle ;
- **la propriété intellectuelle** : créateur, éditeur, collaborateur, droits d'utilisation ;
- **la matérialisation de la ressource** : cycle de vie, type, format, identificateur.

## 4.4 Langages et formalismes

Afin de représenter et d'agencer les données, des langages et formalismes se sont développés. Les plus fréquemment utilisés aujourd'hui sont :

***XML (eXtensible Markup Language)*** a été mis au point en 1996 sous l'égide du W3C[44] (World Wide Web Consortium). C'est un langage structuré de représentation de données pour un document. Plus précisément, c'est un métalangage permettant de rendre explicite la structure des données pour participer à l'interopérabilité entre des données ou des applications.

Un document XML est composé d'un prologue et d'un corps. Le prologue d'un document XML regroupe les métadonnées portant sur le document. On y trouve en particulier la version d'XML, mais aussi éventuellement une représentation formelle de la grammaire du document sous forme directe ou par référence à un fichier externe. Les deux formats de représentation de grammaire aujourd'hui utilisés sont les DTD* (Document Type Definition) qui ont une syntaxe propre, et les schémas dont la syntaxe est exprimée en XML.

Le corps d'un document XML est constitué d'une imbrication de balises délimitant les éléments. Par exemple : *<Protein_Name> Alkane 1-monooxygenase 1</Protein_Name>*.

[43] http://dublincore.org/
[44] http://www.w3.org/

De plus, un élément peut avoir des attributs qui sont utilisés pour représenter à la fois des propriétés et des relations. Cela permet de passer d'une structure hiérarchique d'éléments à une structure en graphe.

Un document XML dont la syntaxe est conforme aux principes précédents est un document bien formé. De plus si la structure de ses éléments est conforme à la grammaire définie ou référencée dans le prologue, le document est dit valide.

XML est donc bien adapté pour décrire explicitement la structure d'un document, il assure une interopérabilité syntaxique. Il faut donc se tourner vers des surcouches de XML, c'est-à-dire des éléments à la structure et au sens bien définis pour représenter la dimension sémantique.

***RDF[45] (Resource Description Framework)***, est un autre standard proposé par le W3C pour la description des sources sur le Web. Les descriptions se font en exprimant des propriétés et en leur attribuant des valeurs. Les schémas RDF, notés RDFS[46], servent à définir les termes et les relations qui interviennent dans ces descriptions.

RDF a pour but de faciliter pour une communauté d'utilisateurs *l'échange* des métadonnées pour des ressources Web partagées et de permettre le traitement de ces métadonnées par des opérateurs humains ou par des machines (proposant des mécanismes de raisonnement automatique). RDF est en effet l'un des modèles de base sur lesquels le Web sémantique* se construit. Le Web sémantique a pour objectif, à plus long terme, d'offrir la possibilité de développer un système d'agents logiciels capables de raisonner en accédant à des ressources variées. Dans ce contexte, le Web sémantique doit d'abord être une infrastructure dans laquelle l'intégration des informations de sources multiples peut être réalisée et facilitée.

Le pouvoir sémantique de RDF se limite à la représentation de la structure de ces concepts, sans parvenir à rendre compte du sens qu'ils véhiculent. Ceci est le rôle des ontologies.

***OWL[47] (Web Ontology Language)*** (Lacot, 2005) est le standard actuellement proposé par le W3C pour représenter les ontologies. Il a été créé pour être utilisé par les applications cherchant à traiter le contenu de l'information et non plus uniquement à présenter l'information. OWL se veut plus représentatif du contenu du Web que XML, RDF et RDF-Schéma en apportant un nouveau vocabulaire avec une sémantique formelle. OWL ajoute du vocabulaire pour décrire les propriétés et classes, comme par exemple la disjonction de classe, la cardinalité (exactement un), l'égalité, les types de propriétés plus riches, les caractéristiques de propriété (symétrie, transitivité, …) et les classes énumérées.

[45] http://www.w3.org/TR/rdf-concepts/
[46] http://www.w3.org/TR/rdf-schema/
[47] http://www.w3.org/TR/2009/WD-owl2-primer-20090611/

OWL est décliné en trois sous langages d'expressivité croissante : OWL lite, OWL DL, OWL Full. OWL Lite est fait pour des besoins préliminaires permettant de définir une hiérarchie et des contraintes simples. Il permet de définir facilement des thésaurus ou taxonomies. OWL DL et Full reposent sur OWL Lite auquel sont ajoutés des constructeurs supplémentaires. OWL DL supporte des besoins d'expressivité maximaux tout en garantissant une complétude de calculs et de décidabilité nécessaires aux systèmes de raisonnement. Il repose sur les éléments OWL auxquels il associe un grand nombre de restrictions (par exemple, une classe peut être une sous-classe de nombreuses autres classes, mais pas une instance d'une classe). OWL DL est conçu pour pouvoir supporter la logique de description. Cette logique appartient à un domaine de recherche qui a pour but d'aider au raisonnement sur une base de connaissances. OWL Full permet un maximum d'expressivité avec la liberté de syntaxe d'RDF. Il n'impose pas de séparation entre classe, propriété, individu et valeur des données. Il permet donc d'augmenter le sens du vocabulaire prédéfini (en OWL ou RDF). Il lève les contraintes imposées par OWL DL pour rendre certaines valeurs disponibles et utilisables dans des bases de données ou de connaissances, mais il ne supporte pas les raisonnements liés à la logique de description.

# CHAPITRE 2

# Approches d'intégration de données en bioinformatique

# Chapitre 2

# Approches d'intégration de données en bioinformatique

**Sommaire**



## 1 INTRODUCTION

Depuis que la navigation manuelle au sein des sources ne suffit plus à résoudre les questions complexes que se posent aujourd'hui par les biologistes, de nombreuses solutions au problème de l'intégration des sources de données ont été proposées. Des systèmes d'intégration ont été développés pour fournir un accès unique via une même interface à plusieurs sources de données, tout en palliant au problème de leur hétérogénéité. Ces systèmes suivent différentes approches, qui varient sur différents points(Hernandez and Kambhampati, 2004).

Trois grandes approches pour l'intégration de sources d'informations ont alors été proposées : *les approches bases de données fédérées, entrepôt et médiateur*.

Dans l'approches *bases de données fédérées*, les sources sont indépendantes les unes des autres et des connections entre toutes les paires de sources que l'on souhaite faire communiquer sont établies. Cette approche est très simple mais très coûteuse puisque permettre à *n* sources de communiquer chacune avec *n-1* sources implique donc d'écrire *n(n-1)* ensembles de connections entre les sources pour supporter les requêtes entre les systèmes (Sheth and Larson, 1990).

L'approche *entrepôt* consiste à voir cette intégration comme la construction d'une base de données réelles, appelée entrepôt, regroupant les informations pertinentes pour les applications considérées. L'utilisateur pose alors ses requêtes ou lance un traitement directement sur les données stockées dans l'entrepôt (Inmon, 1996).

L'approche *médiateur* (Wiederhold, 1992) consiste à fonder l'intégration d'informations sur l'exploitation de vues abstraites décrivant le contenu des différentes sources d'information. Les données ne sont pas stockées au niveau du médiateur et ne sont accessibles qu'au niveau des sources d'information. L'intégration et la détermination des sources d'information pertinentes nécessitent (le plus souvent) la construction de plans de requêtes dont l'exécution permettra d'obtenir l'ensemble des réponses à partir des sources disponibles.

Les approches médiatrice et entrepôt de données demeurent aujourd'hui très répondues. Ces ainsi qu'une grande partie des solutions informatiques pour les données biologiques s'est naturellement orientée vers ces deux architectures. D'autres architectures, comme les portails ou les plateformes, ne cherchant pas (seulement) à intégrer les données mais plutôt à faire interopérer les sources (en utilisant des outils) se sont développées dans le même temps.

Dans ce chapitre, nous allons commencer par présenter les points de variation entre les différentes approches d'intégration, puis nous exposerons l'approche virtuelle suivie de l'approche matérialisée en discutant l'adéquation de chaque solution d'intégration pour les données biologiques. Dans le cadre de Davidson (Davidson, et al., 1995), ces approches sont classées comme intégrant 'fortement' les données. Nous verrons néanmoins que la 'force' d'intégration de ces approches peut varier selon les communautés.

Notre objectif est de montrer la diversité des approches existantes sans chercher à être exhaustifs.

# 2 POINTS DE VARIATION ENTRE LES APPROCHES D'INTEGRATION

On distingue les différentes approches d'intégration selon plusieurs critères que sont : (1) le degré d'intégration, (2) la méthodologie de construction du système d'intégration, (3) la matérialisation des résultats de l'intégration, et (4) les points d'accès aux données (Balko, et al., 2004).

**Le degré d'intégration** est décrit comme étant serré ou lâche. Un système est dit 'à couplage serré' si tous les schémas des sources de données intégrées sont transformés en un modèle de données commun avec le développement d'un schéma global. Un système est considéré comme étant 'à couplage lâche', si un mappage dans un modèle commun a été effectué sans exigence d'aucun schéma global. **La méthodologie de construire un système d'intégration** dépend à plusieurs points : le modèle de données utilisé, les types d'intégration sémantique pris en compte et les méthodes de construction suivies. **La matérialisation des résultats** distingue des solutions matérialisées et autres basées sur les vues. **Les points d'accès aux données** caractérisent la manière d'expression de requêtes envoyées au système.

## 2.1 Degré d'intégration

Principalement il y a deux grandes approches pour l'intégration de données, communément appelées 'approche à couplage serré et approche à couplage lâche'. Selon la première approche, l'intégration des données se réalise par le développement d'un schéma intégrateur contrairement à la deuxième approche qui ne fournit aucun schéma. L'approche à couplage lâche exige un langage de requête unique pour interroger le contenu des sources de données. Ainsi, l'approche à couplage serré offre un schéma, un langage et une transparence d'interface.

### 2.1.1 Approche à couplage serré

Dans le cas de l'approche à couplage serré, qui est souvent mis en œuvre par le biais de l'entrepôt de données, les données sont extraites à partir de sources dispersés dans un seul emplacement physique en fournissant un schéma unifié (schéma intégrateur). Ce schéma peut couvrir l'ensemble des données des sources ou uniquement une partie, mais doit conserver la sémantique des sources de données pour ensuite permettre la pertinence des requêtes. Pour assurer l'équivalence sémantique avec les sources de données et le système d'intégration, il faut établir des correspondances entre le schéma intégrateur et les schémas

des sources. Ces correspondances peuvent être exprimées par des ontologies ou des définitions de règles (voir la sous-section 3.2.1.3).

L'approche à couplage serré a l'avantage d'éviter à l'utilisateur de devoir connaître tous les schémas des sources, mais plutôt d'avoir une connaissance unique du schéma intégrateur.

### 2.1.2 Approche à couplage lâche

Dans l'approche à couplage lâche, aucun schéma n'est nécessaire pour l'interrogation du système. L'approche fournit un langage de requête uniforme qui masque l'hétérogénéité des sources de données où l'utilisateur gère cette hétérogénéité via ses requêtes. Pour faciliter l'accès aux données, ce type de système fournit généralement des vues intégrées. Les utilisateurs peuvent en effets définir des vues sur certaines données qui peuvent ensuite être accessibles pour des requêtes.

Le principal critère pour discerner les deux approches, c'est la visibilité ou non pour les utilisateurs des schémas de sources. Dans l'approche à couplage serré, les schémas de sources ne sont jamais visibles contrairement à l'approche à couplage lâche où les schémas sont toujours visibles.

## 2.2 Méthodologie de développement des systèmes d'intégration

L'intégration sémantique de données regroupe les processus par lesquels les données provenant de différentes sources d'information sont déplacées, combinées et consolidées. Dans ce contexte, le Web sémantique doit d'abord être une infrastructure dans laquelle l'intégration des informations d'une variété de sources peut être réalisée et facilitée. Le Web sémantique devrait donc suivre des méthodes de développement pour la réalisation d'une telle infrastructure.

### 2.2.1 Modèle de données du système d'intégration

L'intégration sémantique est fondée sur la construction d'un modèle de données. Le modèle de données est le schéma global intégrateur (une DTD, un schéma XML, un schéma relationnel…) dans le cas d'une intégration à couplage serré. Il vise à convertir les données des sources en termes de données dans ce schéma global intégrateur. Dans le cas d'une intégration lâche, le modèle de données se base sur le langage de requête utilisé pour accéder aux sources de données.

### 2.2.2 Types d'intégrations sémantique

Certains systèmes intègrent des sources de données complémentaires ne présentant pas d'objets équivalents et exportent donc certaines parties des schémas de celles-ci. D'autres systèmes, au contraire, intègrent des sources de données ayant des contenus chevauchants. Une agrégation* d'information est alors requise pour identifier des objets équivalents d'un point de vue sémantique, c'est-à-dire décrivant le même concept. L'intégration sémantique comporte alors à son tour deux niveaux d'intégrations (différemment qualifiés selon les communautés) : *intégration au niveau des instances* et *intégration au niveau du schéma* ou intégration verticale et horizontale dans la communauté biologique (Hernandez and Kambhampati, 2004; Walter, 2001)), ou encore intégration extensionnelle et intensionnelle (dans la communauté informatique)

**L'intégration au niveau du schéma** vise à intégrer les données en créant une correspondance entre le schéma de chaque source de données et celui du système d'intégration.

**L'intégration au niveau des instances** vise à intégrer les données en identifiant la présence de mêmes objets dans les sources de données. Où on distingue différents niveaux d'intégration sémantique selon que les données sont (1) collectées, sans aucune recherche d'équivalence parmi les objets issus des différents sources ou (2) fusionnées afin d'identifier des objets provenant de sources différentes mais équivalents d'un point de vue sémantique ou (3) supplémentées si les données supplémentaires à celles déjà intégrées viennent décrire le contenu ou la sémantique des données déjà intégrées, on parle alors de métadonnées sémantique.

### 2.2.3 Approches ascendante et descendante

Il existe plusieurs approches pour mettre en place un système d'intégration. Par contre seulement deux approches sont communes (Sen and Sinha, 2005). Il s'agit de l'approche '*top-down*' prônée par Inmon (Inmon, 2002) et l'approche '*Bottom-up*' de Kimball (Kimball, 2002).

**L'approche descendante '*top-down*'** est composée de trois étapes: la collecte des besoins des utilisateurs, la spécification et la formalisation de ces besoins suivant un modèle de données en constellation qui intègre l'expression de contraintes sémantiques. Dans l'approche descendante, les données des sources ne sont pas prises en compte car ces méthodes considèrent que l'objectif d'un modèle de données est de répondre aux besoins des utilisateurs. Elle se base uniquement sur la spécification de ces besoins pour définir les sujets et les axes de l'analyse en négligeant la structure et le contenu des sources à partir desquelles les données décisionnelles sont extraites.

**L'approche ascendante '*Bottom-up*'** fondée sur les données où elle collecte les données à partir des sources de données en se basant sur les schémas de sources, ensuit elle construit un modèle de données pour l'aide à la décision suivant un processus semi-automatique. Autrement dit, La méthode ascendante utilise les sources de données pour définir les besoins des utilisateurs et pour concevoir le schéma du système. Cette méthode considère que les informations pertinentes, pour la prise de décision, se trouvent dans la source (List, et al., 2002).

### 2.3 Matérialisation des résultats

Certains systèmes suivent une approche virtuelle ou non matérialisée. L'approche virtuelle désigne une vision globale, par l'intermédiaire d'un unique schéma de représentation, de l'ensemble des différentes sources de données hétérogènes. Ce schéma global peut être défini automatiquement à l'aide d'outils, ou extracteurs de schéma. Dans cette approche virtuelle les requêtes utilisateurs sont formulées selon la sémantique du schéma global extrait. L'exécution de ces requêtes nécessite une traduction de celles- ci, en sous-requêtes adaptées à chacun des sous-schémas des différentes sources de données.

Certains systèmes au contraire, suivent une approche matérialisée. Dans cette approche, les données, issues de sources hétérogènes, sont stockées localement. Ce stockage permet à l'utilisateur final d'avoir un accès unique et transparent à toutes les données hétérogènes. L'approche matérialisée repose sur une copie des données dans un entrepôt, ainsi les actions sur le référentiel sont asynchrones par rapport aux sources de données. La propagation des modifications apportées au référentiel, vers les différentes sources de données, doit passer par des procédures de mises à jour.

### 2.4 Accès aux données

Un utilisateur accède aux données du système d'intégration selon différentes méthodes pouvant être soit un langage de requête de type SQL ou OQL, soit par le biais de la navigation, spécialement dans les systèmes basées sur le Web.

## 3 APPROCHES D'INTEGRATION EN BIOINFORMATIQUE

Depuis quelques années, de nombreuses solutions au problème de l'hétérogénéité des sources biologiques et à leur intégration ont été proposées. Comme nous avons déjà cité dans la section 2.3, certains systèmes suivent une approche 'non matérialisée' ou une

approche 'virtuelle' dans laquelle les données restent au niveau des sources de données. L'approche virtuelle inclue l'approche de médiation et l'approche navigationnelle. D'autres suivent une approche 'matérialisée' dans laquelle les données sont extraites des différentes sources et combinées dans un schéma global.

## 3.1 Approche non matérialisée

Dans l'approche 'non matérialisée', on distingue tout d'abord des portails, dans lesquels sont regroupés, au sein d'un même site Web, l'accès à diverses banques. Ainsi, les banques de données du NCBI sont actuellement toutes accessibles par le portail *Entrez*[48]. De même, *ExPASy*[49] (Expert Protein Analysis System) (Gasteiger, et al., 2003), construit autour d'Uniprot, est un portail vers un ensemble de sources protéomiques. Certains sites Web proposent un accès unifié et convivial à un ensemble de données complémentaires. SRS[50] (Sequence Retrieval System) (Zdobnov, et al., 2002) (de l'EBI) est un portail qui semble évoluer aujourd'hui vers un réel système d'intégration. Il est basé sur un modèle objet et permet d'interroger 400 banques biologiques de façon uniforme par mots clés. L'originalité de ce portail vient du fait qu'il propose à ses utilisateurs de naviguer à travers les bases comme dans un réseau, en combinant les index des sites des bases et en exploitant leurs références croisées. Ainsi, *GeneCards* (Rebhan, et al., 1997) regroupe un ensemble d'informations permettant une vue générale de la connaissance sur les gènes du génome humain.

Dans les sous-sections suivantes, nous décrivons d'une manière globale deux types d'approches non matérialisées utilisées dans le domaine de l'intégration de données biologiques, le système médiateur et le système navigationnel.

### 3.1.1 Le système médiateur

Dans cette section, nous décrivons l'approche médiateur qui propose de construire un système d'interrogation de sources de données sans toucher aux données qui restent stockées dans leurs sources d'origine. Dans la communauté biologique, l'architecture médiateur est souvent considérée comme une approche bases de données fédérées. Nous indiquerons dans cette section comment certaines approches médiateur sont directement issues des bases de données fédérées. La définition que nous utiliserons d'un médiateur est celle qui est la plus répondue en informatique.

[48] http://www.ncbi.nlm.nih.gov/sites/gquery
[49] http://expasy.org/
[50] http://srs.ebi.ac.uk/

## A) Définition et Architecture

Le médiateur (Wiederhold, 1992) consiste à définir une interface entre l'utilisateur qui pose une requête et l'ensemble des sources accessibles via le Web potentiellement pertinentes pour répondre. L'objectif est de donner l'impression d'interroger un système centralisé et homogène alors que les sources interrogées sont réparties, autonomes et hétérogènes.

Un médiateur (Figure 1) comprend un schéma global, ou ontologie, dont le rôle est central. C'est un modèle du domaine d'application du système. Le schéma global fournit un vocabulaire structuré servant de support à l'expression des requêtes. Par ailleurs, elle établit une connexion entre les différentes sources accessibles. En effet, dans cette approche, l'intégration d'information est fondée sur l'exploitation de vues abstraites décrivant de façon homogène et uniforme le contenu des sources d'information dans les termes de l'ontologie. Les sources d'information pertinents, pour répondre à une requête, sont calculées par réécriture de la requête en termes de ces vues. Le problème consiste à trouver une requête qui, selon le choix de conception du médiateur, est équivalente ou implique logiquement, la requête de l'utilisateur mais n'utilise que des vues. Les réponses à la requête posée sont ensuite obtenues en évaluant les réécritures de cette requête sur les extensions des vues.

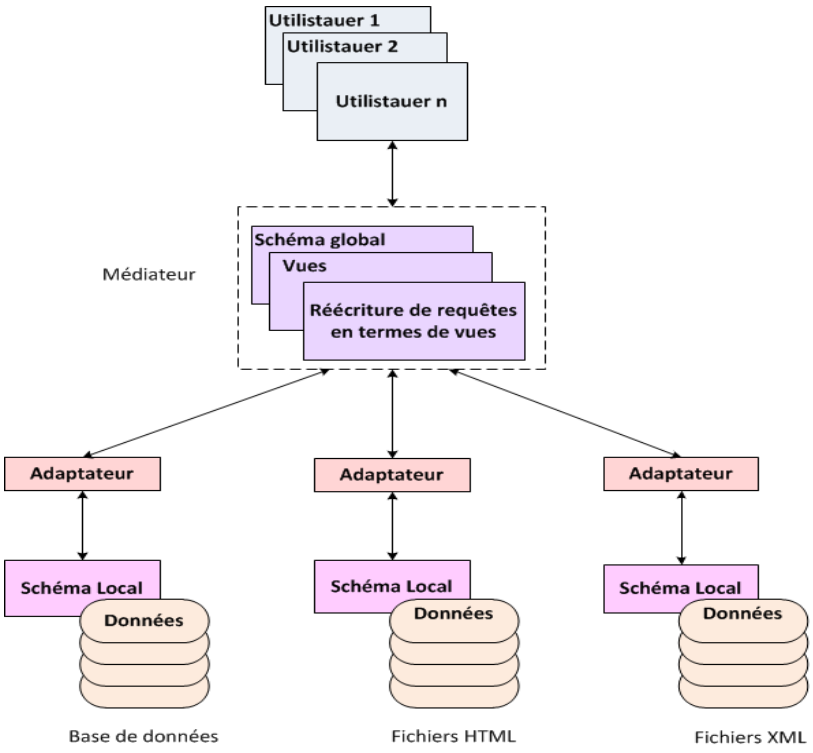


Figure 1. Architecture d'un système médiateur

L'approche médiateur présente l'intérêt de pouvoir construire un système d'interrogation de sources de données sans toucher aux données qui restent stockées dans leurs sources d'origine. Ainsi, le médiateur ne peut pas évaluer directement les requêtes qui lui sont posées car il ne contient pas de données, ces dernières étant stockées de façon distribuée dans des sources indépendantes. L'interrogation effective des sources se fait via des adaptateurs, appelés des wrappers en anglais, qui traduisent les requêtes réécrites en terme de vues dans le langage de requêtes spécifique accepté par chaque source.

### B) Approches GAV, LAV et GLAV

Les différents systèmes d'intégration d'informations à base de médiateur se distinguent par la façon dont est établie la correspondance entre le schéma global et les schémas des sources de données à intégrer (Levy, 1999). On distingue en effet deux manière principales d'établir la correspondance entre le schéma global et les schémas des sources de données à intégrer (GAV et LAV) et une troisième manière qui combine les deux précédentes (GLAV) (Baader, et al., 2003).

**L'approche *Global As View* (GAV)** a été la première à être proposée pour l'intégration d'informations et provient du monde des bases de données fédérées. Elle consiste à définir le schéma global en fonction des schémas des sources de données à intégrer. Pour cela, les structures du schéma global, aussi appelées *relations globales*, sont définies à partir des vues sur les structures des schémas des sources à intégrer. Cette approche alors suppose que les sources à intégrer soient connues à l'avance.

Comme les requêtes d'un utilisateur s'expriment en termes des structures du schéma global, on obtient facilement une requête en termes des schémas des sources de données intégrées, en remplaçant les structures du schéma global par leur définition : on dit que l'on procède au dépliement de la requête. Cette opération de dépliement est effectuée par chainage arrière[51] lorsque les requêtes et les vues sont définies par des règles. Une fois dépliée, une requête peut alors être évaluée de façon standard sur les extensions des sources de données. Ainsi, la construction de la réponse à une requête dans une approche GAV se ramène à l'évaluation standard d'une requête, une fois sa reformulation par dépliement effectuée. L'inconvénient de l'approche GAV est qu'elle est peu adaptée à l'ajout de nouvelles sources de données.

La Figure 2 illustre l'approche GAV où un schéma global *G(A;R:B,C; S:B)* est généré en résumant les schéma sources *R* et *S*. Tous les éléments dans les schémas sources ont des noms correspondants dans le schéma global même si quelques-uns d'entre eux, tels que *R.B* et *S.B*, partagent le même sens. Cependant, il devient difficile de mettre à jour le schéma global à cause de la dépendance entre le schéma global et les schémas locaux. Par

[51] Le mécanisme de chaînage arrière consiste à partir du fait que l'on souhaite établir, à rechercher toutes les règles qui concluent sur ce fait, à construire la liste des faits qu'il suffit de prouver pour qu'elles puissent se déclencher puis à appliquer récursivement le même mécanisme aux faits contenus dans ces listes.

exemple, si le schéma global a été mis à jour (par exemple de nouveaux éléments ont été ajoutés) tous les schémas sources doivent mettre à jour leur vue locale sur le schéma global. D'autre part, l'ajout ou la suppression de sources peut résulter en des modifications considérables sur le schéma global. Comme illustré dans la Figure 2, si un nouveau nœud *T* a été ajouté au système, le schéma global doit être modifié en *G'(A;R:B;C; S:B; T:A;D).*

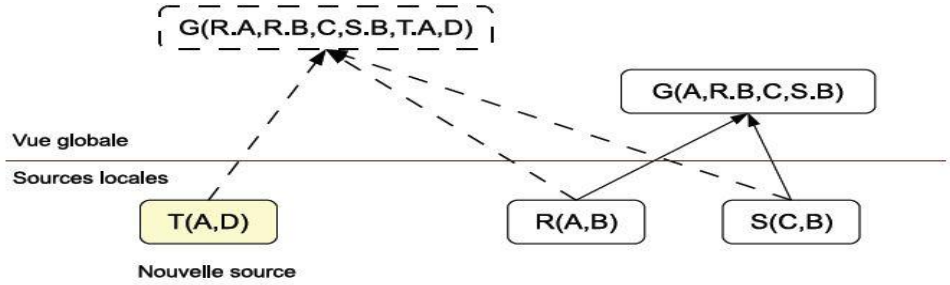


Figure 2. L'approche GAV (Global As View)

**L'approche Local As View (LAV)** est l'approche duale qui consiste à définir les schémas des sources de données à intégrer, en fonction du schéma global. Les avantages et inconvénients de cette approche sont inversés par rapport à l'approche GAV. L'approche LAV (Figure 3) est très flexible par rapport à l'ajout (ou la suppression) de sources de données à intégrer : cela n'a aucun effet sur le schéma global, seules des vues doivent être ajoutées (ou supprimées). En effet, rajouter une source revient à la décrire en fonction du schéma global qui n'est donc absolument pas modifié. Le prix à payer pour cette flexibilité et cette simplicité de mise à jour est la complexité de la construction des réponses à une requête dans un médiateur conçu selon l'approche LAV. La réécriture de requêtes en termes de vues est en effet bien plus complexe que dans une approche GAV. Nous renvoyons le lecteur à (Levy, 1999) pour une discussion formelle.

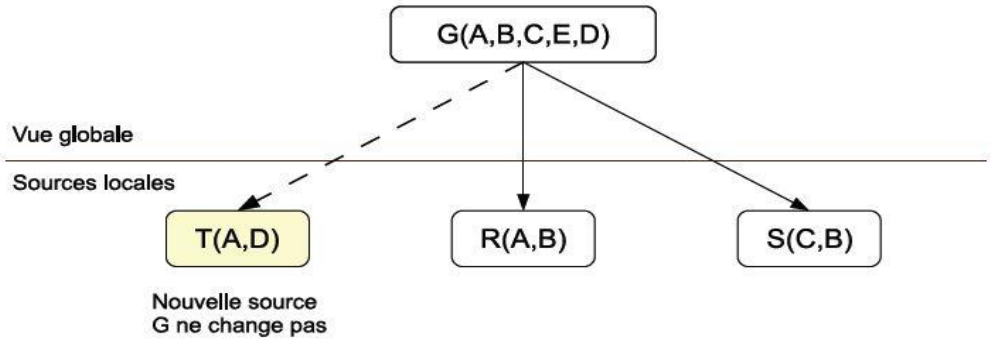


Figure 3. L'approche LAV (Loacl As View)

**Une approche mixte, appelée GLAV** (Baader, et al., 2003), Dans l'approche GLAV (Figure 4), l'intégration entre le schéma médiateur et les schémas locaux est réalisée en combinant les pouvoirs d'expression des approches GAV et LAV. Dans l'approche GLAV, l'indépendance du schéma global, la maintenance nécessaire pour ajouter une nouvelle source et la complexité de la reformulation des requêtes sont les mêmes que dans l'approche LAV. Cependant, GLAV peut créer une vue sur les sources en générant une vue sur le schéma global décrite par les descriptions des sources. Par conséquent, GLAV peut dériver des données en utilisant les vues sur les schémas sources, ce qui est plus expressif que LAV. D'autre part, il permet la reformulation sur le schéma global, ce qui va au-delà du pouvoir d'expression de GAV. On peut remarquer que G' dans la Figure 4 est juste la conjonction de G et du schéma du nouveau nœud T. **La table 1** montre une comparaison entre les trois approches.

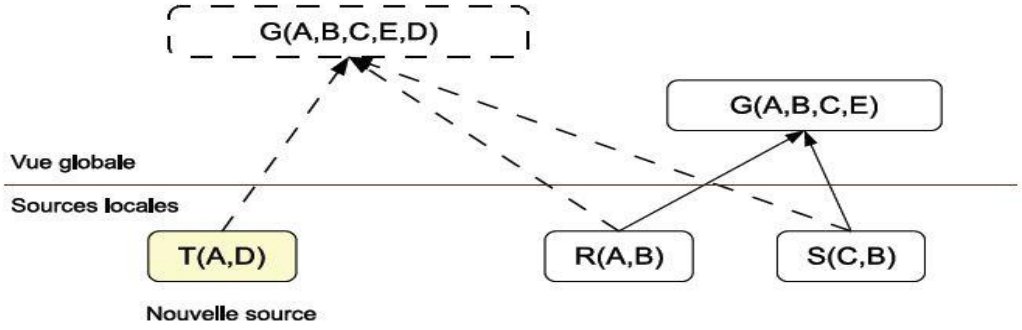


Figure 4. Approche GLAV

**Table1: Comparaison des approches GAV, LAV et GLAV**

| Approche | Réécriture de requête | mise-à-jour source |
|---|---|---|
| GAV | facile | difficile |
| LAV | difficile | facile |
| GLAV | difficile | facile |

## C) Adéquation, Problèmes rencontrés

### (1) Adéquation

L'avantage d'une architecture médiateur est que l'utilisateur n'a pas à se soucier du choix des sources, ce qui est autant plus important qu'il a un grand nombre de sources disponibles sur le Web. D'autre part, l'ajout d'une nouvelle source de données est simple, surtout avec l'approche LAV puisqu'il suffit de décrire la source à ajouter en termes du schéma médiateur. Un médiateur évite toute gestion des mises à jour des données puisque

les données restent dans les sources. Dans le contexte des données biologiques qui évoluent très rapidement cet avantage n'est pas négligeable.

### (2) Problème rencontrés

Quelques problèmes peuvent être rencontrés dans un système médiateur, liés au fait que les données ne sont pas accessibles localement. Le premier est celui du cas de panne d'une source de données. Dans telle situation, on ne peut plus répondre à certaines requêtes.

Le second inconvénient de l'approche médiateur est celui du temps de réponse. Les réponses étant construites à la volée et au fur et à mesure de la collecte des informations au niveau de différentes sources de données. Le temps de réponse à une requête est nettement supérieur à celui qu'on a dans une approche matérialisé où l'interrogation de données se fait directement au niveau des données centralisées.

Grosso modo, les principales difficultés rencontrées dans la construction d'un médiateur sont :

- **Le choix du langage** utilisé pour exprimer le schéma global, ainsi que le choix des langages pour exprimer, en fonction de ce schéma, les vues sur les sources à intégrer et les requêtes des utilisateurs.

- En fonction de ces choix, la conception et la mise en œuvre **d'algorithmes de réécriture** de requêtes en termes de vues pour le calcul des plans de requêtes à exécuter afin d'obtenir l'ensemble des réponses à une requêtes globale.

- **L'évaluation des plans de requête** sur les sources : lors d'une évaluation de plans de requêtes sur les sources, on récupère un ensemble d'instances qui peuvent être potentiellement redondantes. Pour faire correspondre les instances entre elles, il faut suivre les techniques de l'alignement (mappings en anglais).

## D) Panorama des médiateurs existants en Bioinformatique

### (1) K2/Kleisli

Le système K2 (Davidson, et al., 2001), initialement BioKleisli (Davidson, et al., 1997) a été développé à l'université de Pennsylvanie, il est l'un des premiers systèmes de médiation à avoir vu le jour en bioinformatique.

Le médiateur de BioKleisli repose sur un langage de requête de haut niveau, plus expressif que le SQL et qui permet d'interroger plusieurs sources : le CPL (Collection Programming Language) (Hart, et al., 1994). En effet, le langage CPL permet de décomposer une requête complexe en sous-requêtes qui vont être distribuées aux sources concernées par le biais d'adaptateurs. Le système permet d'interroger autant de sources

qu'il intègre d'adaptateurs. Ainsi, il intègre les données sur les voies métaboliques de KEGG (Kyoto Encyclopedia of Genes and Genomes) (Kanehisa and Goto, 2000) et EcoCyc ( Encyclopedia of Escherichia coli) (Keseler, et al., 2005), sur les séquences nucléiques de GenBank et de dbEST[52] (Expressed Sequences Tags databases)(Boguski, et al., 1993), des données spécifiques d'organismes de MGD et de GDB[53] (Human Genome Databases) (Fasman, et al., 1994), des données issues de la recherche de similarités de séquence en utilisant BLAST (Altschul, et al., 1990) et l'ensemble des données indexées par SRS (Sequence Retrival System) (Zdobnov, et al., 2002). BioKleisli est basé sur un schéma orienté objet.

Dans K2, la nouvelle version de BioKleisli, le langage CPL a été remplacé par OQL, un langage plus couramment utiliser car plus proche de la syntaxe du SQL. Un autre aspect intéressant de K2 est la possibilité pour l'utilisateur de définir des vues sur les données non seulement par le biais de requêtes OQL, mais également par la création de nouvelle classes objets. C'est le langage K2MDL (K2 Mediator Definition Language), combinaison du langage ODL (Object Definition Language) et de la syntaxe OQL qui permet à l'utilisateur de créer de nouvelles classes en spécifiant comment leurs attributs sont instanciés par les sources de données. Ces nouvelles vues peuvent ensuite être interrogées par OQL.

**(2) TAMBIS**

Tambis (Transparent Access to Multiple Bioinformatic Information Sources) est un système de médiation basé sur une ontologie développé à l'université de Manchester (Stevens, et al., 2000). L'originalité du système est d'être basé sur une ontologie TaO (Tambis Ontology) (Baker, et al., 1999). Les requêtes dans TAMBIS sont formulées à travers une interface graphique où l'utilisateur navigue à travers les concepts définis au niveau du schéma global et choisi ceux qui l'intéressent pour la requête courante. Le système utilise la logique de description GRAIL (Rector, et al., 1997), qui est aussi utilisée pour exprimer des requêtes sur le système. Les concepts sont organisés en hiérarchie, et les rôles assurent des relations binaires entre concepts. Ainsi, lorsqu'un utilisateur pose une requête, il explore l'ontologie et choisit la combinaison de concepts et de rôles nécessaire à la formulation de sa requête. Cette requête est ensuite convertie dans le langage CPL de K2, assurant la transformation des requêtes pour adopter les paramètres propres aux sources de données. Le résultat du plan de requêtes est ensuite délivré à l'utilisateur au format HTML.

TaO a été ensuite écrit avec le langage DAML +OIL (Stevens, et al., 2002), puis avec OWL qui sont des langages plus expressifs.

[52] http://www.ncbi.nlm.nih.gov/dbEST/
[53] http://gdbwww.gdb.org/

Ainsi, TAMBIS fournit un accès transparent aux sources de données où l'utilisateur n'a besoin ni de connaître les sources à interroger pour une requête donnée, ni être familier avec un langage de requête particulier.

**(3) DiscoveryLink**

DiscoveryLink est projet d'IBM résulte de la fusion de Garlic[54] (Roth, et al., 1996) et de DataJoiner (Gupta and Lin, 1994) (qui est basé sur DB2 (Chamberlin, 1998)). Il utilise une architecture de médiation et des adaptateurs afin de proposer une couche intermédiaire d'accès aux données de plusieurs sources biologiques. DiscoveryLink (Haas, et al., 2001) utilise le modèle de données relationnel-objet ; il résout les problèmes d'hétérogénéité syntaxique, mais ne prend pas en compte les différences sémantiques. Les requêtes sont soumises en SQL sur le schéma global, un plan d'exécution est généré puis optimisé; l'utilisateur n'a pas à se préoccuper des sources locales, dont l'accès est géré par les adaptateurs. DiscoveryLink a désormais changé son nom en Information Integrator (Arenson, 2003), mais fonctionne toujours selon le même principe.

**(4) BACIIS**

Le projet BACIIS (Biological And Chemical Information Integration System) est un système de médiation qui intègre des données biologiques et chimiques. Comme TAMBIS, BACIIS est fondé sur une ontologie sous-tendue par une logique de description. La logique de BACIIS est Loom (MacGregor R and Bates R, 1987) qui est moins expressive que le langage GRAIL mais aussi moins coûteuse. L'ontologie de BACIIS (BAO) a trois dimensions : les classes (hiérarchie classique, is-a), les propriétés (attributs des classes, organisés en hiérarchies) et les relations (liens entre les classes). Certaines métadonnées (liées aux références croisées entre les sources) et les problèmes de traçabilité ne sont que rapidement évoqués dans la publication (Mahoui, et al., 2005).

La particularité de BACIIS est l'intégration d'un plus grand nombre de sources de données. Les concepteurs du système considèrent en effet que l'intégration de sources de données chevauchantes, par exemple deux banques de données protéiques, permet d'obtenir des résultats plus pertinents. En effet, BACIIS fournit des solutions au problème d'absence de données dans certaines sources, et de conflits entre données dus aux inconsistances dans les sources de données. Ceci est effectué par une évaluation de la correspondance sémantique entre deux objets de sources différentes. Un algorithme permet d'éliminer les données sémantiquement distantes dans le processus d'intégration.

[54] http://www.almaden.ibm.com/cs/garlic/

### 3.1.2 Le système navigationnel

Cette approche s'inspire de ce que font habituellement les utilisateurs lors d'une recherche d'information sur le Web, qui implique une recherche de page en page par clic de souris. Elle ne nécessite aucun apprentissage particulier d'un langage de requêtes dédié et permet de choisir les sources à utiliser. Le schéma global présenté à l'utilisateur est facile à construire, car il se contente d'unir ceux des sources entre eux. Les données des banques sont ensuite intégrées en se basant sur leurs références croisées. En pratique, les requêtes sont générées à partir de formulaires sur le Web, dont les paramétrages choisis sont transformés en expressions de chemin. C'est une approche intéressante puisqu'elle permettre d'accéder à des informations uniquement accessibles via une navigation entre les sources de données (Friedman, et al., 1999). Les résultats fournis par une première requête peuvent être utilisés comme point de départ pour de nouvelles interrogations.

*A) Définition*

L'approche navigationnelle ne sous-entend pas une modélisation des données elles-mêmes mais plutôt une modélisation représentant les sources comme un ensemble de pages avec des interconnections et des points d'entrée, ainsi que des informations complémentaires telles que la spécification du contenu des sources, des éventuelles contraintes de chemins, et des paramètres facultatifs et obligatoires d'entrée (Hernandez and Kambhampati, 2004).

Notons que comparé au nombre important de sources de données actuellement disponibles sur le Web, nombre qui a atteint 1380 selon les critères de Michael Galperin dans son référencement publié chaque années dans le journal Nucleic Acids Research (Galperin and Fernández-Suárez, 2012), le nombre de références croisées est faible. Les sources les plus importantes partagent des identifiants, mais nombreuses sont celles, plus petites, qui soit adoptent un système d'identification propriétaire, soit ne proposent que partiellement des références partagées. Les systèmes basés sur le partage de références souffrent d'un manque de flexibilité lors de l'ajout d'une source ; le calcul de toutes les interconnexions fait surgir le problème $N^2$ (Morris, 2003). L'intégration navigationnelle atteint donc rapidement ses limites lorsque le nombre de sources qui intéressent l'utilisateur augmente, et peut mener à des problèmes de désorientation et de surcharge cognitive (Martin, 1996). L'expression des vues et des jointures est difficile, puisque souvent limitée par le manque d'expressivité inhérent aux formulaires de requêtes utilisés sur internet. Malgré ses défauts, l'intégration navigationnelle peut avoir des avantages pour interroger rapidement des sources hétérogènes et distribuées et confronter leurs informations. Elle ne nécessite pas d'apprentissage, et se présente comme un moyen simple d'accélérer ce qui est fait encore aujourd'hui manuellement.

*B) Exploitation des références croisées*

Les liens entre les données génomiques sont de natures variées, On peut distinguer dans un premier temps les liens qui conduisent à des données sur une même entité (par exemple,

Protéine à Protéine, de UniProt à Protein du NCBI) des liens qui apportent des informations sur une autre entité (par exemple, Gène à Pathologie de GenBank à OMIM[55]).

Ensuite, on distingue les liens internes permettant d'accéder à des données d'une même source (par exemple, KEGG vers KEGG) des liens externes permettant d'accéder à des données d'une autre source (par exemple GenBank vers AmiGO[56]). Les liens externes sont également qualifiés de références croisées, ou cross-références, ils ne sont pas nécessairement symétriques. Il y a par exemple un grand nombre de sources qui cross-référencent GenBank et qui ne sont pas référencées en retour.

La plupart de sources de données font référence à des informations communes sur lesquelles il est possible de s'appuyer afin de rassembler les données. Les liens que nous considérons se basent sur la présence d'une entité commune entre deux sources, comme le montre l'exemple de la Figure 5.

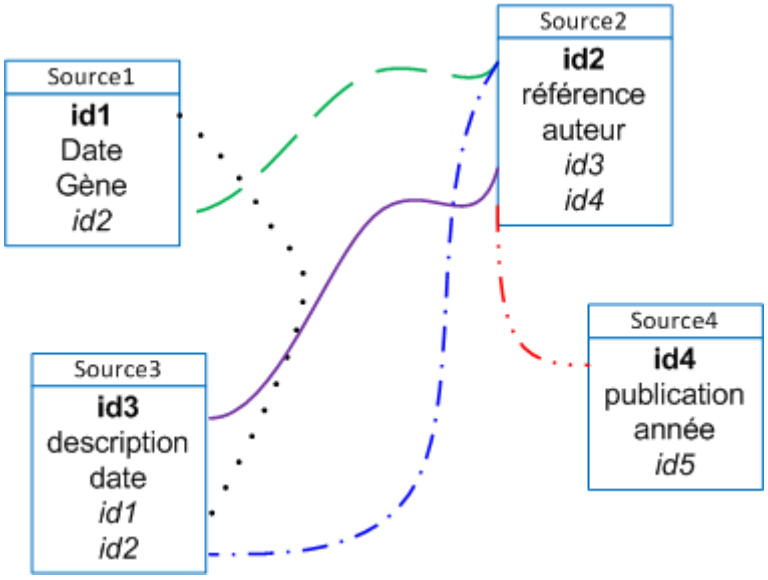


**Figure 5. Exemple de partage de références entre les sources**

Regardons en détail les brève descriptions des quatre sources présentées dans l'exemple de Figure 5; nous voyons que chacune possède un identifiant unique (numéro d'accession pour certains bases de données) pour les données qu'elle contient (indiqué en gras), mais aussi des références aux identifiants des autres sources (indiquées en italique). Sur notre exemple illustratif, plusieurs chemins peuvent être empruntés pour obtenir les mêmes données. Supposons par exemple que l'utilisateur souhaite intégrer la *description* la *référence* et l'*identifiant* d'un gène à partir de la données *date* de découverte qu'il connaît.

[55] http://www.ncbi.nlm.nih.gov/omim
[56] http://amigo.geneontology.org/cgi-bin/amigo/go.cgi

La Figure 6 illustre le graphe de liens existants entre les quatre sources pour répondre à la requête.

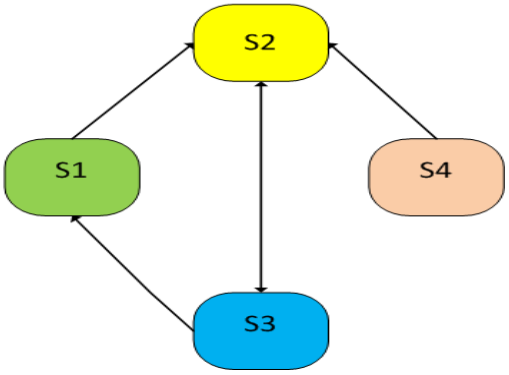


Figure 6. Graphe de liens entre les sources

En utilisant les sources *Source1*, *Source2* et *Source3* ; deux possibilité se représentent :

- Soit en interrogeant *Source1*, puis *Source2* grâce à id2, et enfin *Source3* grâce à id3
- Soit en interrogeant d'abord *source3*, pour ensuite réutiliser les identifiants qu'elle possède afin d'interroger *Source1* et *Source2*

**La table 2** synthétise les deux scénarios possibles. La collecte s'arrête dès qu'une boucle apparaît dans le parcours des sources.

**Table 2 : Les deux déroulements possibles**

| **Collecte de données entre S1, S2 et S3 à partir d'une date** | |
|---|---|
| **Scénario 1** | **Scénario 2** |
| Requête avec une date sur S1<br>⬇<br>Requête sur S2<br>À partir de id2 tiré de S1<br>⬇<br>Requête sur S3 | Requête avec une date sur S3<br>⬇<br>Requête sur S1 et S2<br>A partir de id1 et id2 tirés de S3 |

Cet exemple simple nous a permis de mettre en évidence qu'il existe plusieurs chemins possible pour obtenir les données souhaitées.

Dans certain nombre de cas, il est impossible de satisfaire la requête de l'utilisateur simplement à partir des sources qu'il a choisi. Sur notre exemple précédent, ce cas de figure apparaît si on souhait extraire les publications de la *Source4* associées à des gènes extraits de la *Source1*. Il est impossible de joindre ces données sans passer par une source intermédiaire. La *source2* doit être utilisée alors qu'elle ne fait pas partie du choix de l'utilisateur, et qu'elle n'apporte aucune information supplémentaire.

L'exploitation des références partagées entre les sources biologiques afin d'intégrer les données a déjà été le centre de plusieurs projets. Ces projets sont discutés dans la sous-section suivante.

### *C) Panorama des systèmes navigationnels existants en Bioinformatique*

Les systèmes développés utilisant l'approche navigationnelle varient en fonction de plusieurs critères. On constate différents niveaux de transparence laissés à l'utilisateur pour le choix des sources à interroger, une prise en compte ou non des différents chemins traversant les sources pouvant être générés pour une même requête, et la manière dont sont évalués ces différents chemins.

#### (1) Le système SRS

SRS (Sequence Retrieval System) est un système qui a été initialement développé par l'EMBL puis par l'EBI afin de faciliter l'accès aux banques de séquences (Etzold and Argos, 1993; Etzold, et al., 1996). Depuis 1999, SRS est valorisé et commercialisé par LION Bioscience AG[57]. Il permet d'interroger à l'aide d'une même interface, 400 banques de données (Zdobnov, et al., 2002).

SRS est plus un système de recherche par mot clé qu'un véritable système d'intégration. En effet, son approche d'intégration repose sur l'utilisation du langage de description et d'exploration des données ICARUS (Interpreter of Commands And Recursive Syntax) qui permet d'indexer toute source de données structurée. Ce langage est d'abord utilisé pour parcourir les sources de données structurées afin d'identifier les données qui y sont décrites puis créer des index pour chacune de ces données. Ces index sont stockés localement et sont utilisés lors des interrogations pour la recherche d'informations. Même si ces index sont stockés localement, SRS ne constitue pas un entrepôt de données puisque les données elles-mêmes ne sont pas intégrées.

Ainsi, le principal avantage de ce système est la possibilité de pouvoir indexer en même temps une grande quantité de banques sans se soucier de l'organisation de celles-ci et

[57] http://www.biochipnet.com/node/1561

donc de pouvoir manipuler avec le même langage les principales banques généralistes et beaucoup de banques spécialisées.

ICARUS autorise la création automatique d'un réseau de cross-références, permettant ainsi la navigation inter-banques. Cette fonctionnalité fait qu'il est possible de relier entre elles des collections ne présentant pas directement de cross-références.

La formulation de requêtes via SRS se fait par l'intermédiaire d'une interface Web. SRS propose aux utilisateurs de choisir la source de données à interroger, ainsi que le mot clé ou la séquence à rechercher. Plusieurs critères de sélection ou plusieurs sources peuvent être utilisés par le biais d'opérateurs logiques ET, OU et NON. SRS délivre le résultat de la recherche ainsi que toute information relative à la requête en exploitant le réseau de cross-références. L'utilisateur peut ainsi accéder (par simples clics) à des informations complémentaires contenues dans d'autres sources.

Si SRS utilise les cross-références présentes dans les sources de données biologiques pour satisfaire au mieux les requêtes, ce système n'offre aucune transparence au niveau des sources, et n'exploite en aucun cas la diversité de chemins pouvant être générée pour une même requête.

### (2) Le système BioMediator

Le système BioMediator, initialement GeneSeek (Mork, et al., 2001), a été développé à l'université de Washington. Les concepteurs de BioMediator optent pour un niveau de transparence où l'utilisateur dépose une requête au système, puis récupère son ou ses résultats sans avoir à spécifier les chemins à parcourir et donc les sources à interroger. Plusieurs chemins peuvent être parcourus pour répondre à une même requête, et l'ensemble des résultats par chemin est délivré à l'utilisateur.

Le système BioMediator suit une conception modulaire, composé de six composant (Figure 7) qui effectuent l'intégration des données sur plusieurs sources de données biologiques structurés et semi-structurées.

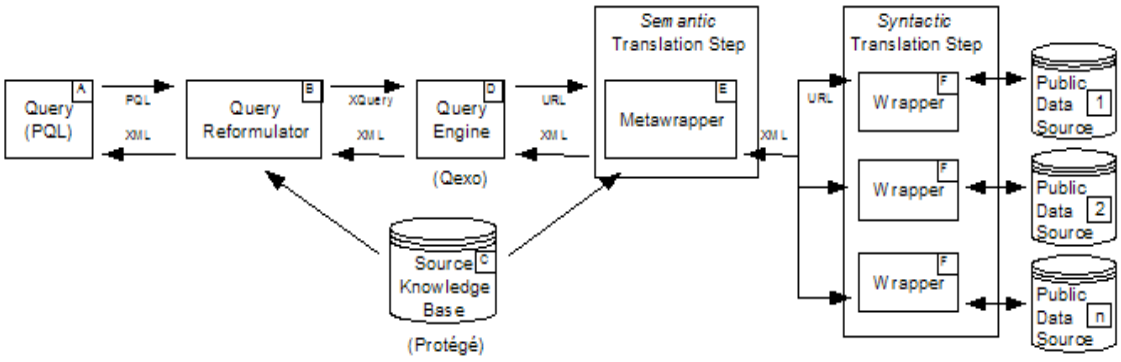


Figure 7. Diagramme d' architecture de BioMediator adapté de

Dans un sens large, le système BioMediator définit et traverse un graphe où les nœuds représentent des instances de sources de données pour les entités dans le schéma de médiation. Les arêtes représentent des instances des relations qui relient les entités entre une ou plusieurs sources et le schéma. Lors d'une exécution, un chemin entre deux entités d'intérêt peut être construit par la concaténation de plusieurs arêtes au niveau graphe.

PQL (Figure 7 A) (Mork, et al., 2002) est un langage de requête basé sur le chemin, PQL contient des règles permettant à l'utilisateur de spécifier des contraintes de la requête et le chemin entre les bases de données. Le Reformulator (Figure 7 B) accepte les requêtes d'entrée PQL et énumère tous les chemins. La base de connaissances de la source (SKB) (Mork, et al., 2001) (Figure 7 C) est représenté par Protégé58 et est accessibles via l'API Protégé. Elle contient: a) toutes les entités, les attributs et les relations dans le schéma médiation, b) le catalogue de toutes les sources de données possibles et les éléments de schéma médiation qu'ils contiennent, c) les règles de mappage pour une translation sémantique et bidimensionnelle des flux entre les requêtes et les sources de données (Shaker, et al., 2002). Le moteur d'exécution de requête (Qexo[59], (Figure 7 D)) accepte XQuery comme entrée et des URLs* comme sortie. Le metawrapper (Shaker, et al., 2002) (Figure 7 E)transforme les URLs en requêtes effectuées sur les sources par l'application des règles de mapping stockées au niveau de SKB. Finalement, les adaptateurs envoient les requêtes aux spécifiques sources de données. Les résultats consistent en un ou plusieurs chemins, ainsi que les données retrouvées par ces différents chemins.

Mork et al. ont au départ cherché à déterminer la validité des différents chemins (Mork, et al., 2001). Pour ce faire, ils ont utilisé comme critère, la cardinalité des références, qui correspond au nombre d'entrées retrouvées par une référence, et ont attribué une confiance d'autant plus haute que la cardinalité était réduite (Mork, et al., 2002). Par la suite, Mork et al. ont préféré que l'évaluation des « bons chemins » soit faite par l'utilisateur plutôt que par le système lui-même. Ainsi, avec PQL, le système délivre l'ensemble des chemins possibles, plutôt qu'une liste réduite.

**(3) Le système BioNavigation**

BioNavigation est un système d'intégration également basé sur l'approche navigationnelle. Il a été développé à l'université d'Arizona (Lacroix, et al., 2005a).

Ce système utilise les ontologies afin d'éviter à l'utilisateur, lors d'une interrogation, d'avoir à spécifier les sources à utiliser. D'après Lacroix, ceci permet aux utilisateurs de ne pas restreindre leurs requêtes aux caractéristiques et aux limitations des sources qu'ils ont l'habitude d'utiliser. Ainsi, BioNavigation utilise deux niveaux de représentation : le niveau physique qui décrit les sources, leurs contenus et leurs liens entre elles, et le niveau logique

[58] http://protege.stanford.edu/
[59] http://www.xml.com/pub/a/2003/06/11/qexo.html

ou « ontologie BioNavigation » qui décrit les entités biologiques, les relations entre ces entités ainsi que les correspondances avec les sources contenant ces entités (Figure 8).

L'ontologie permet à l'utilisateur de visualiser et de naviguer au sein des différentes entités biologiques et ainsi de sélectionner graphiquement celles qui sont nécessaires à la construction d'une requête (Lacroix, et al., 2005b). Un utilisateur souhaitant récupérer les citations discutant d'un gène particulier va d'abord graphiquement sélectionner l'entité 'Gène' puis la relation 'discuté dans' puis l'entité 'Citation'.

BioNavigation fournit à l'utilisateur l'ensemble des chemins possibles pour une requête donnée. Mais BioNavigation apporte une plus-value en fournissant à l'utilisateur des moyens pour évaluer et optimiser les choix de chemins.

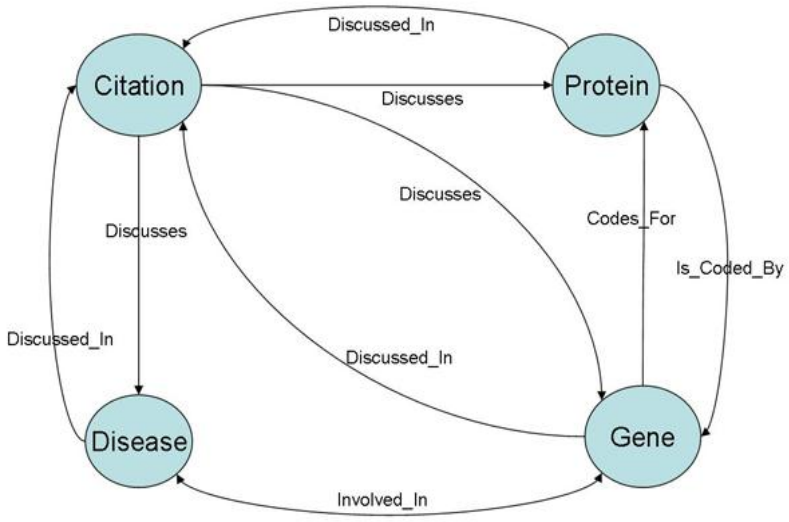


Figure 8. Exemple de graphe d'entités (Niveau logique)

Les concepteurs du système BioNavigation ont en effet démontré qu'en fonction du choix du chemin, différents facteurs peuvent varier comme le coût en temps d'exécution de la requête, la qualité et la quantité des résultats obtenus (Lacroix and Edupuganti, 2004). Toutefois, ils avancent qu'il n'y a pas un seul « meilleur chemin » pour répondre à une requête, mais plutôt plusieurs meilleurs chemins puisque plusieurs paramètres peuvent permettre d'évaluer la satisfaction d'un chemin. Ainsi, dans BioNavigation, lors de l'exécution d'une requête, tous les chemins possibles sont générés et sont classés selon trois paramètres :

**La cardinalité du chemin** : C'est le nombre d'instances de chemins du résultat. Pour un chemin de longueur 1 entre deux sources S1 et S2, c'est le nombre de paires liées (e1,e2), où e1 est une entrée de S1 et e2 de S2.

**La cardinalité de la cible** : C'est le nombre d'objets retrouvés dans la source finale.

**Le coût de l'évaluation** : C'est le coût total de la requête incluant le coût d'exécution locale et les délais d'accession aux sources.

Le classement ainsi obtenu permet à l'utilisateur de sélectionner le chemin qui le satisfait au mieux en fonction de ses besoins. En effet, la cardinalité du chemin reflète la probabilité qu'il existe un chemin entre deux sources, la cardinalité de la cible indique le nombre de résultats en sortie et le coût de l'évaluation guide l'utilisateur dans le choix du chemin le plus efficace en temps.

### (4) Le système BioGuide

Les concepteurs du système de BioGuide ont apporté une dimension nouvelle à l'approche navigationnelle, il s'agit de la prise en compte des notions de préférence et de stratégies des utilisateurs (Cohen-Boulakia, et al., 2004) (Cohen-Boulakia, et al., 2005). En effet, BioGuide, un système qui aide l'utilisateur à choisir des sources pertinentes et des outils bioinformatiques adaptés à sa requête. BioGuide offre un réel support dans le processus d'interrogation en proposant une représentation sous forme de graphe (a) du domaine biologique (entités biologiques et relations entres elles) et (b) du réseau formé par les outils et les références croisées présents entre les sources. L'utilisateur peut interagir avec ces graphes et peut également les modifier s'il le souhaite. Il peut exprimer sa requête en y sélectionnant des éléments (les entités pour lesquelles il recherche de l'information, le type de sources à consulter...). En retour, BioGuide lui fournit la liste des sources à consulter et des outils à utiliser ainsi que l'ordre dans lequel il doit considérer ces sources et outils, sous la forme de chemins entre les sources. Ces chemins sont construits en respectant les préférences de l'utilisateur et en suivant la stratégie de son choix.

**Les préférences** : Les enquêtes ont permis d'identifier 30 critères déterminant la préférence des utilisateurs, et permettant donc de filtrer et/ou de classer les chemins générés pour une requête donnée. Parmi ces critères citons la fiabilité et la facilité d'utilisation.

**Les stratégies** : De manière naturelle un utilisateur souhaitant accéder au résultat d'une requête impliquant plusieurs sources, va naviguer au travers les sources pour lier les différentes entités biologiques impliquées dans la requête. Mais il existe des différences de stratégies selon si oui ou non les utilisateurs i) suivent un ordre dans le parcours des entités au sein des sources, ii) explorent des entités intermédiaires à celles contenues dans la requête et iii) visitent une source donnée une seule fois.

Globalement, BioGuide suit des étapes de (I) à (IV) (Figure 9): (I) la *requête initiale de l'utilisateur* $Q$ se compose de (i) $Q_{entRel,}$ les entités et les relations sémantiques de la requête ; et (ii) les choix de l'utilisateur sur les critères choisis de stratégies (*ordre et entités-seulement*). (II) À partir de $Q$, le module *EPG* génère *ENTITY PATHS*, l'ensemble des chemins dans le graphe des entités construit selon les critères de stratégie *ordre* et *entités-seulement*. (III) La *requête raffinée de l'utilisateur* $Q_{se}$ (ayant pour support le graphe des sources-entités) se compose de (a) *ENTITY PATHS*, la sortie du module *EPG*, (b) le choix de l'utilisateur sur le critère de stratégie *source-une-fois-pour-toutes*, et (c) les préférences de l'utilisateur. (IV) À partir de $Q_{se}$ et du graphe des sources-entités, le module *SEPT* génère la liste *PATHS* des chemins de sources-entités qui peuvent être utilisés pour récolter des données.

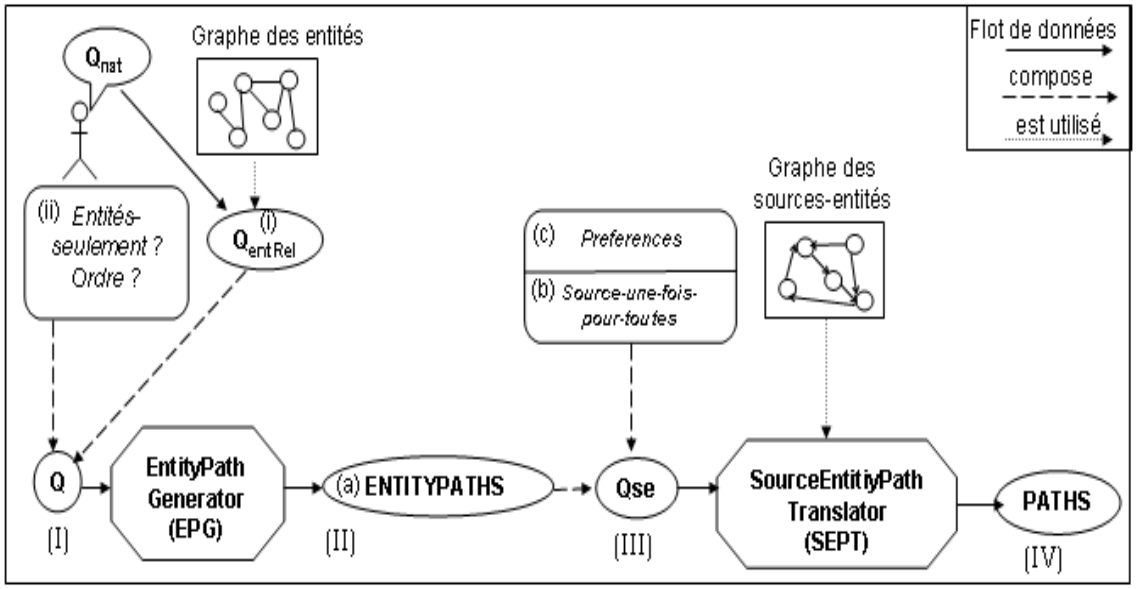


**Figure 9. Architecture de BioGuide**

Le système BioGuide fournit une interface permettant à un utilisateur de formuler ses propres requêtes, mais également de régler ses propres paramètres de préférences et de stratégies. Un utilisateur peut ainsi filtrer sur différents niveaux : les chemins, les entités ou les sources. Il peut ensuite combiner différentes stratégies. Les concepteurs de BioGuide ont démontré qu'une telle approche permet non seulement de rassembler un plus grand nombre d'informations, mais aussi de confronter et donc de comprendre des données divergentes entre chemins différents (Cohen-Boulakia, et al., 2005).

## 3.2 Approche matérialisée (Entrepôt de données)

Construire un entrepôt de données consiste à matérialisé localement les données récupérées sur les sources, les transformer afin de les rendre compatible avec le schéma global préalablement défini, faire la part des redondances et des complémentarités, puis exécuter des requêtes sur les données consolidées. L'entrepôt de données, ou *data warehouse*, est un concept spécifique de l'information décisionnelle, issu du constat suivant : les données de l'informatique de production (également appelée 'informatique transactionnelle') ne se prêtent pas à une exploitation dans un cadre d'analyse décisionnelle. Les systèmes de production sont en effet construits dans le but de traiter des opérations individuelles qui peuvent impliquer différents métiers du laboratoire ou de l'entreprise, et surtout, ne se préoccupent pas de leur compilation ou de leur historisation dans le temps. À l'inverse, les systèmes décisionnels doivent permettre l'analyse par sujets ou par métiers. Il est donc souvent de séparer ces deux mondes et de repenser les schémas de données, ce qui implique l'unification des différents gisements de données en un entrepôt de données global.

### 3.2.1 Définition et Architecture

A) Définition

Le père du concept[60], dans son livre '*Builiding the Data Warehouse*' (Inmon, 2002), décrit l'entrepôt de données : « *l'entrepôt de données est une collection de données orientées sujet, intégrées, non volatiles et historisées, disponibles pour le support d'un processus d'aide à la décision.*» L'entrepôt n'est pas une simple copie des données de production. Il est organisé et structuré, et se caractérise par des données que nous les détaillons selon (Franco, 1997) :

- **Orientation sujet** : Les données d'un entrepôt s'organisent par sujets ou thèmes. Cette organisation permet de rassembler toutes les données, pertinentes à un sujet et nécessaires aux besoins d'analyse, dans une structure unique.

- **Intégration** : Les données d'un entrepôt sont le résultat de l'intégration de données en provenance de multiples sources ; ainsi, toutes les données nécessaires pour réaliser une analyse particulière se trouvent dans l'entrepôt. L'intégration est le résultat d'un processus qui peut devenir très complexe due à l'hétérogénéité des sources.

- **Non volatiles** : Une requête lancée à différentes dates, en précisant la date de la référence de l'information recherchée, donnera le même résultat. Les données sont non volatile, elles ne disparaissent pas après les mises à jours.

[60] http://en.wikipedia.org/wiki/Bill_Inmon

- **Historiée** : A la différence des données opérationnelles, celles de l'entrepôt sont permanentes et ne peuvent pas être modifiées. Le rafraîchissement de l'entrepôt, consiste à ajouter de nouvelles données, sans modifier ou perdre celles qui existent. Un référentiel de temps doit alors être associé aux données afin d'identifier les valeurs particulières dans le temps.

- **Disponible pour le support d'un processus d'aide à la décision** : Des outils d'analyse et d'interrogation doivent permettre aux utilisateurs de consulter facilement les données.

### B) Architecture

Dans la Figure 10 nous présentons une architecture simplifiée d'un entrepôt de données en détaillant les différentes couches qui le constituent.

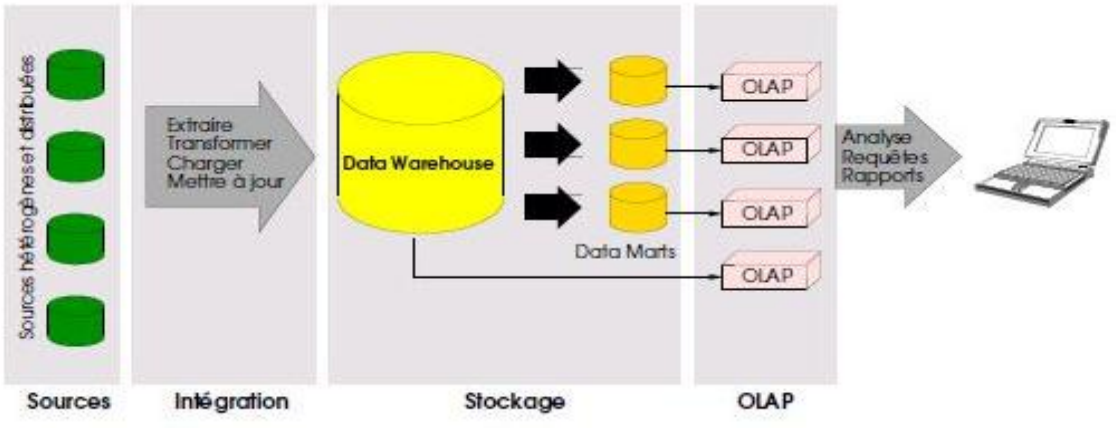


Figure 10. Architecture d'un entrepôt de données

Les données de l'entrepôt sont extraites de diverses sources souvent réparties et hétérogènes, et qui doivent être transformées avant leur stockage dans l'entrepôt. Les Data Marts* sont chargés de répondre aux requêtes émises par les utilisateurs. Ils sont alimentés depuis l'entrepôt de données et interroger par les outils d'analyse de type OLAP* (On Line Analytical Processing) (voir la sous-section 3.2.2).

Les données d'un entrepôt de données se trouvent selon deux axes (Figure 11) : synthétique et historique. L'axe synthétique établie une hiérarchie d'agrégation et comprend les données détaillées (qui représentent les événements les plus récents au bas de la hiérarchie), les données agrégées (qui synthétisent les données détaillées) et les données fortement agrégées (qui synthétisent à un niveau supérieur les données agrégées) (Benitez-

Guerrero, et al., 1999). L'axe historique comprend les données détaillées historisées, qui représentent des événements passées. Les Métadonnées contiennent des informations concernant les données dans l'entrepôt de données, telle que leur provenance et leur structure, ainsi que les méthodes utilisées pour faire l'agrégation.

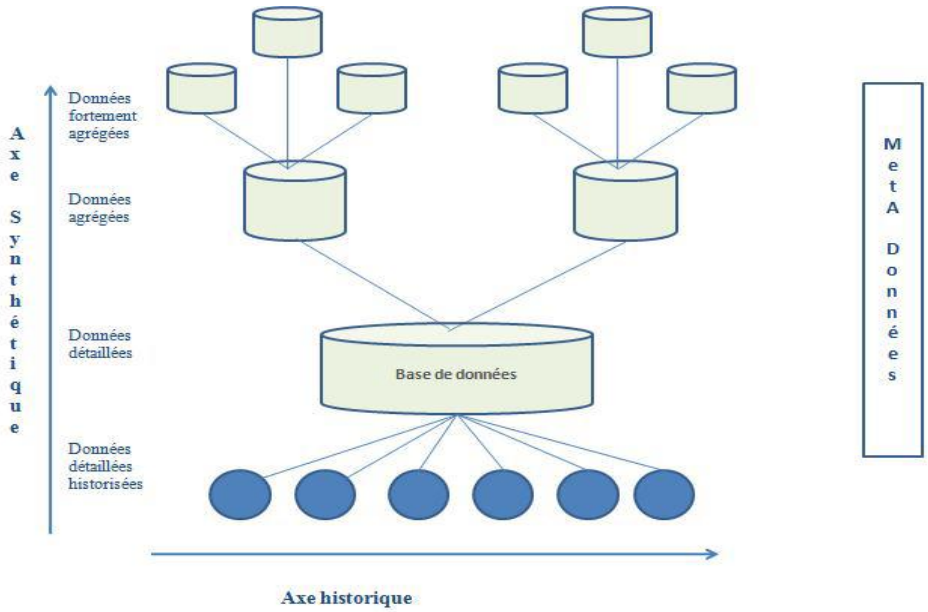


Figure 11. Architecture et niveaux d'agrégation des données

### 3.2.2 Intégration de données dans un système entrepôt

L'intégration est la procédure qui permet de transférer les données des sources externes vers l'entrepôt de données, en les adaptant. Elle est divisée en quatre étapes qui sont : 1) l'extraction des données des sources, 2) la transformation des données aux niveaux structurel et sémantique, 3) l'intégration des données et enfin 4) le stockage des données intégrées dans le système cible.

Il faut noter que cette décomposition est seulement logique. L'étape d'extraction et une partie de l'étape de transformation peuvent être groupées dans le même composant logiciel, tel qu'un adaptateur (wrapper) ou un outil de migration de données. L'étape d'intégration est souvent couplée avec des possibilités de transformation de données dans un même composant logiciel, qui, habituellement, réalise le chargement dans l'entrepôt de données. Toutes les étapes de traitement peuvent aussi être groupées dans un même logiciel. Quand les étapes d'extraction et d'intégration sont séparées, les données nécessitent d'être stockées entre les deux. Ceci peut être fait en utilisant un middleware par source ou un middleware pour toutes les sources.

Une vue opérationnelle typique de ces composants est donnée par la Figure 12.

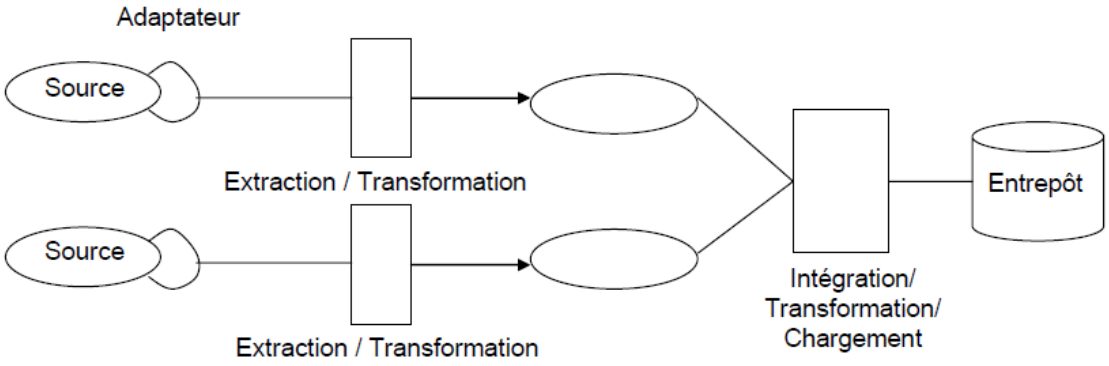


Figure 12. Vue opérationnelle des composants utilisés pour la construction d'entrepôt de données

L'un des principaux problèmes posés par l'intégration des données consiste à effectuer la transformation des données du format des sources vers le format de l'entrepôt de données. Ce processus de transformation requiert la mise en correspondance structurelle et sémantique entre le schéma des sources de données et le schéma global de l'entrepôt de données (Bernstein and Rahm, 2000). Il s'agit de la correspondance inter-schémas ou appariement de schémas (schema matching).

Il existe différentes approches de correspondance inter-schémas. Elles dépendent du type d'information du schéma qui est utilisé et comment cette information est interprétée (Rahm and Bernstein, 2001). Commençons par rappeler les définitions de schéma et de correspondance inter-schémas.

Un schéma est un ensemble d'éléments connectés par une certaine structure. En pratique, il existe différentes représentations comme le modèle relationnel, le modèle orienté objet ou le XML. Dans chacune des représentations, on distingue des éléments et des structures : les entités et les relations dans le modèle relationnel, les objets et les relations dans le modèle orienté objet et les éléments et les sous-éléments dans le XML.

Etant donné un schéma global G et une source de données dont le schéma est noté S, la correspondance inter-schémas consiste à identifier les éléments des deux schémas (S et G) qui se correspondent, et comment ces éléments sont reliés. On distingue différents types de relations entre les éléments de deux schémas. Ils peuvent être directionnels (un élément de S correspond à un élément de G) ou non directionnels (une combinaison d'éléments de S et G se correspondent). Il peut s'agir de relations par le biais d'opérateurs (= ; > …) ou de fonctions (addition, concaténation). Il peut s'agir de relations d'ensembles (chevauchement, contenance) ou toute autre relation exprimée en langage naturel.

L'implémentation des correspondances inter-schémas se fait par des algorithmes, qui se basent sur différents critères pour établir les correspondances. On distingue les critères de classification suivants (Rahm and Bernstein, 2001) :

**Instance versus schéma** : Les correspondances peuvent être effectuées à partir des instances (le contenu des données) ou seulement à partir de l'information contenue au niveau du schéma.

**Elément versus structure** : Les correspondances peuvent être effectuées pour des éléments individuels du schéma ou pour des combinaisons d'éléments, comme des sous-structures complexes de schémas.

**Langage versus contrainte** : Les correspondances peuvent se baser sur des approches linguistiques (en utilisant les noms des éléments du schéma, par exemple égalité de nom, synonymie, etc …) ou sur des approches de contraintes (en utilisant les relations).

**Correspondance de cardinalité** : La correspondance peut être basée sur la relation d'un ou plusieurs éléments d'un schéma avec un ou plusieurs éléments de l'autre schéma, ceci menant à quatre cas : 1:1, 1:n, n:1, n:m.

**Information auxiliaire** : Un certain nombre d'algorithmes de correspondance ne reposent pas uniquement sur les schémas en entrée mais sur des informations auxiliaires, telles que les dictionnaires, les schémas globaux ou des correspondances déjà effectuées.

Il faut noter que certains algorithmes effectuent les correspondances en se basant sur un seul de ces critères, alors que certains combinent plusieurs critères.

### 3.2.3 Système d'information transactionnel versus décisionnel

Le développement de l'entrepôt de données est une conséquence de l'observation, par W. Inmon, au début des années 90, sur le fait que le niveau opérationnel du traitement transactionnel OLTP* (On Line Transactionnel Processing) et les applications d'aide à la décision OLAP (On Line Analytical Processing) ne peuvent pas coexister efficacement dans le même environnement de bases de données, essentiellement à cause de leurs caractéristiques transactionnelles très différentes. L'entrepôt de données est différent des systèmes d'informations classiques qualifiés de Système d'Information transactionnel, car les besoin par lesquelles on veut le construire sont différents (Franco, 1997).

Les systèmes d'information transactionnels sont communément appelés OLTP pour indiquer qu'ils servent à traiter des processus transactionnels en ligne. Ces systèmes sont caractérisés par un nombre d'utilisateurs important, des interrogations et des modifications fréquentes, et des volumes de données par transaction relativement faible. Dans ce cadre, le modèle de données est destiné à minimiser les redondances pour préserver la fiabilité et la cohérence du système. De cette manière le système garantit une

réduction des temps d'exécution et facilite les procédures d'ajout, de suppression et de modification.

À l'inverse, les entrepôts de données sont dédiés à la prise de décision. Ils sont qualifiés de OLAP car l'exploitation des informations contenues dans ces systèmes est réalisée par des processus d'analyse en ligne des données (Codd, et al., 1993). Ces systèmes sont utilisés par un nombre restreint d'utilisateurs et privilégient le fait de pouvoir poser une grande variété de requêtes de manière interactive et plus rapide qu'en OLTP sur de grands volumes de données. Ces requêtes peuvent être simples, ou au contraire plus complexes, permettant ainsi de mettre en relation des éléments qui a *priori* ne sont pas corrélés au départ. Il faut donc une organisation qui permet de mémoriser de grands jeux de données et qui facilite la recherche de connaissance. Ainsi, l'entrepôt de données est entièrement construit selon une approche dimensionnelle. De plus, l'information qu'il contient est mise à jour par des sources de données externes lors de procédures de chargement. Aussi, le modèle de données doit assurer l'intégrité des données lors de l'intégration. Ceci implique une cohérence du schéma global de l'entrepôt et une alimentation réfléchie et planifiée dans le temps.

### 3.2.4 Les modèles des entrepôts de données

La conception d'un entrepôt de données est très diffèrent de celle d'une base de données transactionnelles, puisque les besoins en termes d'analyses sont différents. Un entrepôt de données repose sur un modèle multidimensionnel de données.

A) La modélisation conceptuelle

La conception des bases de données se base en général sur le modèle Entité Association (E-A). Ce modèle permet de décrire des relations entre les données élémentaires (entités) en éliminant les redondances, ce qui provoque l'introduction d'un nombre important de nouvelles entités.

De ce fait, l'accès aux données devient compliqué et le diagramme généré difficile à comprendre pour un utilisateur. C'est pour cette raison que l'utilisateur de la modélisation E-A pour la conception d'un entrepôt n'est pas considéré comme approprié.

**(1) Concept de fait, de dimension et de hiérarchie**

Le modèle multidimensionnel est une alternative mieux adéquate aux besoins de l'analyse des données d'un entrepôt. La modélisation multidimensionnelle part du principe que l'objectif majeur est la vision multidimensionnelle des données. Le constructeur fondamental de ces modèles est le cube de données (Figure 13), qu'offre une abstraction très proche de la façon dont l'analyse voit et interroge les données. Il organise les données

en une ou plusieurs **dimensions***[61], qui déterminent une mesure d'intérêt ou bien le **fait***[62]. Une dimension spécifie la manière dont on regarde les données pour les analyser, alors qu'une mesure est un objet d'analyse. Chaque dimension est formée par un ensemble d'attributs et chaque attribut peut prendre différentes valeurs.

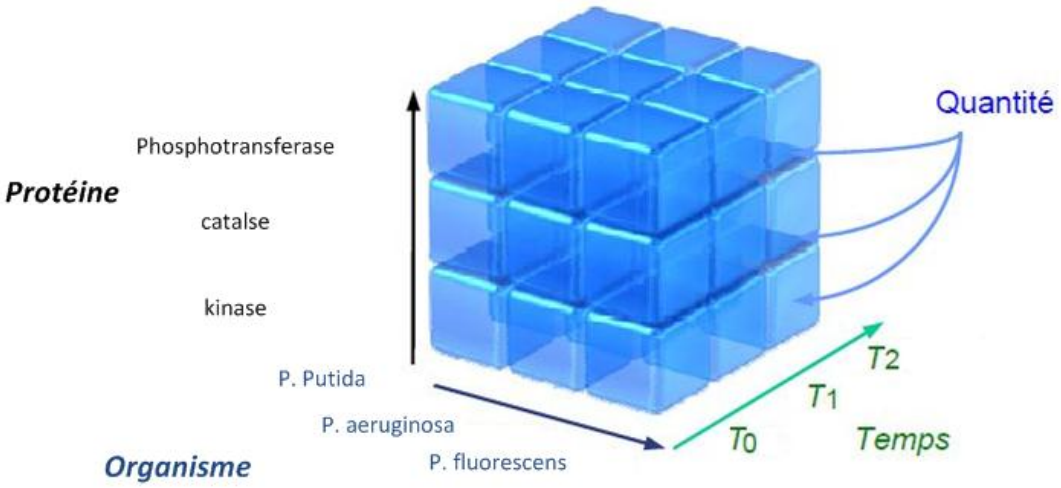


Figure 13. Exemple de cube de données

Les dimensions possèdent en général des hiérarchies associées qui organisent les attributs à différents niveaux pour observer les données à différentes granularités. Une dimension peut avoir plusieurs hiérarchies[63] associées, chacune spécifiant différentes relations d'ordre entre ses attributs.

Dans la Figure 13 , on peut alors observer les données dans un espace à trois dimensions : la dimension Protéine, la dimension Organisme et la dimension Temps. Chaque intersection de ces dimensions représente une cellule comportant la Quantité de la protéine.

### (2) Modèles en étoile, en flocon et en constellation

A partir du fait et des dimensions, il est possible d'établir une structure de données simple qui correspond au besoin de la modélisation multidimensionnelle. Cette structure est constituée du fait central et des dimensions (Figure 14). Ce modèle représente visuellement une étoile, on parle de modèle en étoile.

[61] Une dimension modélise une perspective de l'analyse. Une dimension se compose de paramètres correspondant aux formations faisant varier les mesures de l'activité

[62] Le fait modélise le sujet de l'analyse. Un fait est formé de mesures correspondant aux informations de l'activité analysée

[63] Une hiérarchie organise les paramètres d'une dimension selon un ordre conformément à leur niveau de détail

Le modèle en étoile se compose du fait central et de leurs dimensions. Dans ce schéma il existe une relation pour les faits et plusieurs pour les différentes dimensions autour de la relation centrale. La relation de faits contient les différentes mesures et une clé étrangère pour faire référence à chacune de leurs dimensions.

Il existe d'autres techniques de modélisation multidimensionnelle, notamment la modélisation en flocon (snowflake). Une modélisation en flocon est une extension de la modélisation en étoile, il consiste à garder la même table des faits et à éclater les tables de dimensions afin de permettre une représentation plus explicite de la hiérarchie (Jagadish, et al., 1999). Elle peut être vue comme une normalisation des tables de dimensions. L'avantage du schéma en flocon de neige (Figure 15) est de formaliser une hiérarchie au sein d'une dimension, ce qui peut faciliter l'analyse. Un autre avantage est représenté par la normalisation des dimensions, car nous réduisons leur taille. Néanmoins dans (Kimball, 2002), l'auteur démontre que c'est une perte de temps de normaliser les relations des dimensions dans le but d'économiser l'espace disque. Par contre, cette normalisation rend plus complexe la lisibilité et la gestion dans ce type de schéma. En effet, ce type de schéma augmente le nombre de jointures à réaliser dans l'exécution d'une requête.

Dans l'exemple ci-dessus (Figure 15), la dimension '*Dimension 3*' a été éclatée en trois, '*Dimension 3*', '*Sous-type*' et '*Type*'. La dimension 'Dimension 1' a été décomposé en quatre : '*Dimension 1*', '*Ss-ss-Cat*', '*Sous-Cat*' et '*Catégorie*'.

Le schéma en constellation (Figure 16) fusionne plusieurs modèles en étoile qui utilisent des dimensions communes. Un modèle en constellation comprend donc plusieurs faits et des dimensions communes (Benitez-Guerrero, et al., 2001).

### B) La modélisation logique

Au niveau logique, plusieurs possibilités sont envisageables pour la modélisation multidimensionnelle. Il est possible d'utiliser :

- un système de gestion de bases de données existant tels que les SGBD relationnels (ROLAP*) ou bien les SGBD orientes objet (OOLAP).
- un système de gestion de bases de données multidimensionnelles (MOLAP*).

L'approche la plus couramment utilisée consiste à utiliser un système de gestion de bases de données relationnelles, on parle de l'approche ROLAP (Relational On-Line Analytical Processing). Le modèle multidimensionnel est alors traduit de la manière suivante:

- Chaque fait correspond à une table, appelé **table de fait**,
- Chaque dimension correspond à une table, appelée **table de dimension**.

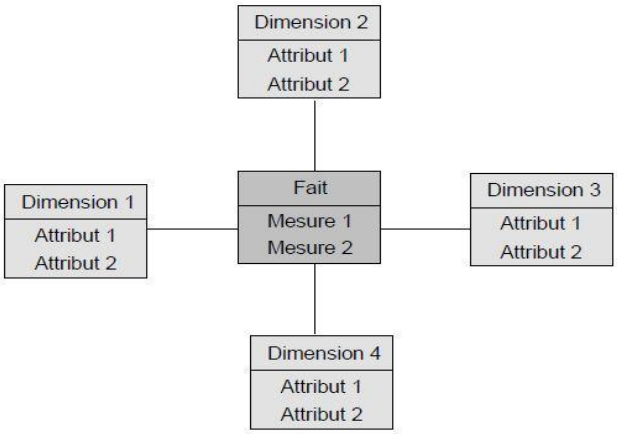


Figure 14. Modèle en étoile

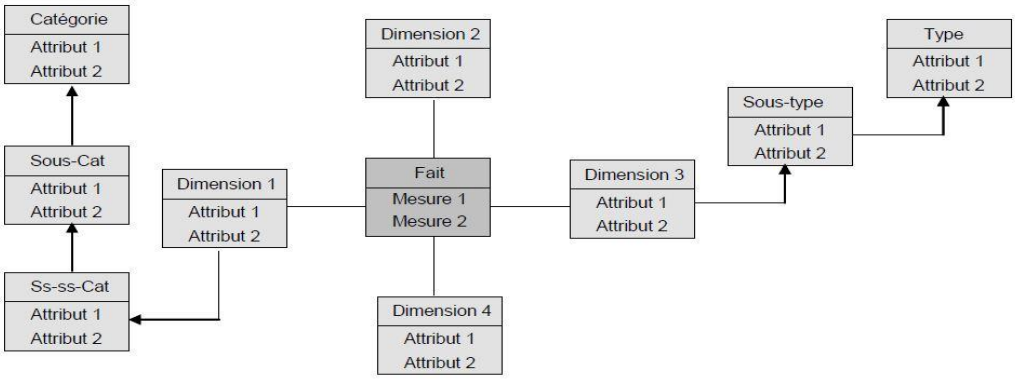


Figure 15. modèle en flocon

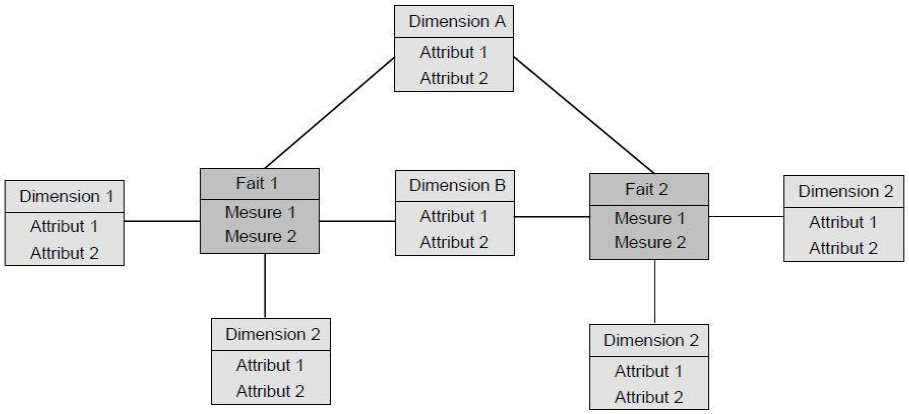


Figure 16. Modèle en constellation

Ainsi, la table de fait est constituée des attributs représentant les mesures d'activités et les attributs clés étrangers de chacune des tables de dimension. Les tables de dimension contiennent les paramètres et une clé primaire permettant de réaliser des jointures avec la table de fait.

Plus récemment, une autre approche s'appuie sur le paradigme objet ; on parle de l'approche OOLAP (Object On-Line Analytical Processing). Le modèle multidimensionnel se traduit ainsi :

- Chaque fait correspond à une classe, appelée **classe de fait**,
- Chaque dimension correspond à une classe, appelée **classe de dimension**.

Pour décrire les expressions qui décrivent le schéma en étoile ou en flocon, on utilise le langage de définition standard des bases de données orientées objet défini par (Object Data Management Group) l'ODMG[64].

Une alternative à ces deux approches consiste à utiliser un système multidimensionnel. Les systèmes de type MOLAP stockent les données dans un SGBD multidimensionnel sous la forme d'un tableau multidimensionnel. Chaque dimension de ce tableau est associée à une dimension du cube. Seules les valeurs de données correspondant aux données de chaque cellule sont stockées (Figure 13). Ces systèmes demandent un pré-calcul de toutes les agrégations possibles. En conséquence, ils sont plus performants que les systèmes traditionnels, mais difficiles à mettre à jour et à gérer.

Les systèmes MOLAP apparaissent comme une solution acceptable pour le stockage et l'analyse d'un entrepôt lorsque la quantité estimée des données d'un entrepôt ne dépasse pas quelques giga-octets. Mais, lorsque les données sont éparses, ces systèmes sont consommateurs d'espace (Chaudhuri and Dayal, 1997) et des techniques de compression doivent être utilisées.

L'intérêt est que les temps d'accès sont optimisés, mais cette approche nécessite de redéfinir des opérations pour manipuler ces structures multidimensionnelles. Parmi les utilisées sont :

*Pivot* : Cette opération consiste à faire effectuer à un cube une rotation autour d'un des trois axes passant par le centre de deux faces opposées, de manière à présenter un ensemble de faces différents.

*Switch* : Cette opération consiste à inter-changer la position des membres d'une dimension.

*Split* : Elle consiste à présenter chaque tranche du cube, et à passer d'une représentation tridimensionnelle d'un cube à sa représentation sous la forme d'un ensemble

[64] www.odmg.org

de tables. D'une manière générale, cette opération permet de réduire le nombre de dimensions d'une représentation. On notera que le nombre de tables résultant d'une opération Split dépend des informations contenues dans le cube de départ et n'est pas connu à l'avance.

### C) La modélisation de données XML multidimensionnelles

L'augmentation de l'échange de données entre applications a incité la création de standards tels que XML, aujourd'hui omniprésent. D'énormes quantités de données sont maintenant disponibles au format XML et les outils permettant d'utiliser ces données s'améliorent chaque jour. Plus particulièrement, les bases de données XML natives et le langage d'interrogation XQuery, sont aujourd'hui suffisamment avancés pour être utilisés dans un environnement de production. L'approche traditionnelle pour l'entreposage de données XML est de les convertir en données relationnelles. Cependant, mettre en place un entrepôt de données utilisant uniquement les technologies XML est une piste de recherche intéressante. Les données peuvent être modélisées en tant que documents XML stockés dans une base de données XML native et analysés à l'aide de requêtes XQuery.

L'approche *X-Warehousing* (Figure 17) (Boussaïd, et al., 2006; Choquet and Boussaïd, 2007) est entièrement basée sur XML. Elle apporte un niveau d'abstraction pertinent pour préparer ces derniers à l'analyse. Elle permet de construire des cubes XML. Ces derniers sont composés chacun d'une collection de documents XML. Chaque document correspond alors à un fait OLAP et doit satisfaire certaines contraintes, comme respecter une information minimale pour que le fait à observer soit consistant. Pour cela, la validation des documents par un schéma XML est une tâche indispensable. Ce dernier représente le modèle conceptuel du cube qui généralement consiste en un schéma en étoile ou en flocons de neige.

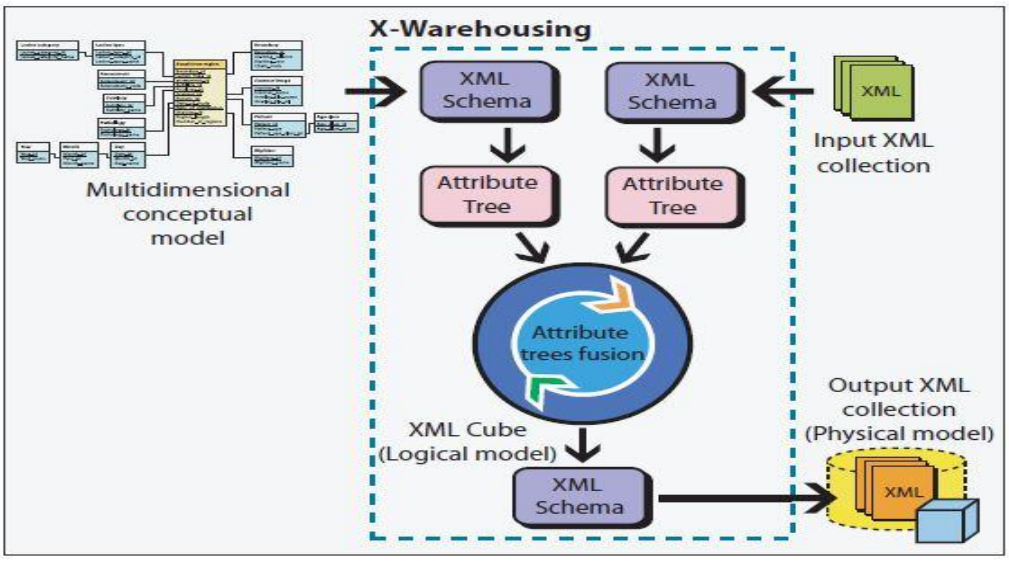


Figure 17. Les étape de l'approche X-Warehousing

La Figure 17 résume les différentes étapes de l'approche *X-Warehousing*, où l'utilisateur déclare ses objectifs d'analyse sous la forme d'un modèle conceptuel multidimensionnel (MCM). Ce modèle est exprimé par un schéma XML puis transformé en un arbre d'attributs également représenté par un schéma XML. La contribution de cette approche est d'obtenir un ensemble homogène de données avec des contraintes strictes sur leurs contenus.

Selon (Boussaïd, et al., 2006), le fait (ou cube) étant défini comme un document XML unique. Chaque document XML de ce cube représente un fait OLAP constitué d'un ou plusieurs indicateurs (mesures) à observer à travers des axes d'analyse (dimensions et hiérarchies de dimensions). L'ensemble des documents XML entreposés correspond au modèle physique du cube de données qui est désigné par cube XML.

### 3.2.5 Adéquation, Problèmes rencontrés

#### (1) Adéquation

Si beaucoup d'entrepôts de données se sont développés dans le secteur commercial depuis les années 90, ce n'est que depuis récemment que l'utilisation de l'approche entrepôt s'est répandue en bioinformatique (Kasprzyk, et al., 2004). Ceci s'explique par le fait que les données biologiques, contrairement aux données de l'entreprise, sont plutôt descriptives et non numériques, et de nature complexes et hétérogènes. Ainsi, les processus de mise en œuvre de l'entrepôt deviennent plus complexes. Cependant, de nombreux avantages de l'approche ont tout de même motivé son utilisation dans le secteur de la bioinformatique (Davidson, et al., 2001; Hernandez and Kambhampati, 2004) :

**La grande capacité de gestion et de stockage** : L'entrepôt de données peut stocker de larges volumes de données. Ceci est très bien adapté à la gestion de données provenant de multiples sources privées et/ou répandues sur le Web, mais également à la gestion de données issues des nouvelles technologies qualifiées de « haut débit ».

**La représentation multidimensionnelle des données** : L'organisation des données par dimension est très adaptée à la manière avec laquelle sont spécialisées par thèmes les sources de données génomiques sur le Web. Cependant, il faut prendre en considération le fait que certaines sources ont des contenus chevauchants. Ainsi, plusieurs sources de données peuvent être utilisées pour représenter une dimension, c'est-à-dire un thème.

**La performance des requêtes** : Les données sont matérialisées physiquement au sein d'un schéma global. Les temps de connexion aux sources de données lors des requêtes sont éliminés, et les requêtes sont optimisées car elles sont exécutées localement.

**La transformation de données lors de l'intégration** : Le processus de transformation des données avant leur intégration dans un schéma global permet de réconcilier les contenus provenant de sources de données chevauchantes (intégration verticale) et/ou complémentaires (intégration horizontale) (voir sous-section 2.2.2). Ce processus permet de résoudre les nombreux problèmes de nomenclature des gènes et de réconcilier cette connaissance au sein d'un même schéma.

**La modification des données par l'utilisateur** : Les données étant disponibles localement, l'utilisateur peut filtrer, valider ou invalider, rectifier ou annoter les données provenant des sources. Ainsi, l'expertise de l'utilisateur peut être prise en compte.

**(2) Problèmes rencontrés**

Les difficultés liées à l'architecture entrepôt se rencontrent d'abord lors de la **construction** de l'entrepôt puis lors de sa **maintenance**. Construire un entrepôt nécessite une étude des sources à intégrer pour identifier les informations pertinentes à stocker puis une extraction des données des sources. On construit alors le **schéma intégrateur**. Selon les cas, cette tâche peut se faire manuellement ou par l'utilisation d'algorithmes (pour la détection d'analogies entre les structures des sources, par exemple). Cette étape nécessite notamment de choisir un langage adapté à la représentation des informations à stocker dans l'entrepôt. **L'insertion des données** dans l'entrepôt est souvent précédée d'une série de nettoyages des données visant à supprimer les redondances possibles et les divergences des données des sources (intégration sémantique au niveau des schémas et des instances).

Maintenir l'entrepôt consiste à **mettre à jour** les copies de l'entrepôt par rapport aux sources, ce qui impose d'élaborer des mécanismes permettant de détecter quand et comment les données des sources changent. Pour ce faire, on développe des algorithmes incrémentaux.

Le problème de la mise à jour des données est accru dans le domaine biologique car les sources évoluent extrêmement vite et n'indiquent pas précisément quelles annotations ont été ajoutées/supprimées/détruites de leurs données mais listent simplement les fiches d'annotations qui ont été touchées par une mise à jour.

### 3.2.6 Panorama des entrepôts de données existants en Bioinformatique

A) GUS

L'entrepôt GUS (Genomics Unified Schema) (Davidson, et al., 2001) est le premier grand entrepôt de données biologiques, et il est encore à l'heure actuelle le plus important. GUS est une plate-forme générique de gestion de données sur les organismes modèles ou sur les maladies. GUS intègre des données très diverses, depuis les données génomiques aux protéomiques en passant par les données transcriptomiques. Il offre en outre un support pour l'annotation semi-automatique, le nettoyage des données, la fouille de données et

l'analyse de requêtes complexes. GUS a un schéma générique. Il est en effet utilisé pour stocker des données diverses : du génome complet « Plasmodb[65] » (Collaborative, 2001) aux données biomédicales liées au pancréas « EPConDB[66] » (Mazzarelli, et al., 2007) .

Le schéma de GUS comporte plus de 180 tables divisées en 5 domaines distincts (provenance des données, ontologies utilisées pour annoter les données, séquences et annotations, données d'expression, données de régulation des gènes). GUS intègre de nombreuses sources, notamment GenBank, UniProt, Prodom, InterPro, GO, dbEST et dbSNP[67]. Le schéma de GUS est constitué de l'union des schémas des sources mais il possède aussi un ensemble de tables fortement intégrées où les données sont le résultat d'une série d'algorithmes qui permettent l'unification des instances. Une sous-partie des données de GUS est donc intégrée au niveau sémantique. C'est là la particularité de GUS : chaque utilisateur peut définir des traitements sur les données de l'entrepôt et choisir de regrouper les entrées de son choix, il contribue ainsi un peu plus à l'intégration verticale.

### B) GEDAW

Gene Expression DAta Warehouse (Guérin, et al., 2005) est un entrepôt de données développé au sein de l'équipe bioinformatique de l'INSERM U522 (Régulations des équilibres fonctionnels du foie normal et pathologique) en collaboration avec l'IRISA de Rennes. Il est spécialisé dans les données du transcriptome hépatique et dédié à l'analyse des données générées par son étude. Ces données sont de natures et d'origines variées, dont une bonne partie se trouve disséminée dans des sources biomédicales sur le Web très disparates (au niveau des contenus et des structures), qu'il faut intégrer. La finalité de GEDAW est de fournir une aide à la décision permettant d'orienter les recherches biologiques. La fouille précise des données expérimentales enrichies par les données intégrées est destinée à émettre des hypothèses qui vont ainsi guider la recherche sur le foie.

GEDAW utilise des techniques d'intégration à partir de sources de données structurées ou semi-structurées uniquement (GenBank au format XML, GeneOntology, UMLS, et le Transcriptome au format relationnel). GEDAW propose des règles de correspondance pour regrouper plusieurs fiches de GenBank qui décrivent une même instance biologique, en l'occurrence un même gène. Ces règles de correspondance peuvent être définies en utilisant des alignements de séquences (si un BLAST entre deux séquences renvoie un fort score de similarité alors les deux séquences sont relatives au même gène), ou encore en utilisant l'inclusion de séquences (la séquence contenue dans une fiche est incluse dans celle contenue dans une autre). Par son expertise, le chercheur biologiste peut lui aussi émettre des règles de nettoyage des données.

[65] http://plasmodb.org/plasmo/
[66] http://www.cbil.upenn.edu/epcondb42/
[67] http://www.ncbi.nlm.nih.gov/projects/SNP/

Dans GEDAW, l'intégration se fait donc au niveau des schémas, essentiellement les schémas de GenBank (définis par des DTDs), mais surtout au niveau des instances elles-mêmes avec une intégration horizontale et verticale. Dans le premier cas, des techniques de détection des analogies structurelles et des correspondances ont été mises en place afin de transformer les structures des sources vers une forme canonique (le schéma global). Dans le second cas, la réconciliation des données se fait par regroupement d'entrées pour identifier les instances. Cette identification se fait donc à l'aide de l'expression de critères pour faire correspondre les entrées et éliminer les redondances et les divergences des informations.

### C) BioWarehouse

BioWarehouse (Lee, et al., 2006) a été conçu et développé comme un système de construction et de gestion d'entrepôts de données, afin de permettre l'interopérabilité de bases de données bioinformatiques disparates. Les sources définies à la conception de BioWarehouse sont : BioCyc[68], CMR[69], GenBank, KEGG et Uniprot.

L'extraction des données s'effectue selon la lecture des bases définies et le chargement de données est fait dans la base de BioWareHouse selon le schéma global de l'entrepôt (conversion des sources en un schéma relationnel et selon la sémantique de BioWarehouse). Chaque module de chargement (loader) est spécifique à la source correspondante, ces modules sont implémentés généralement en C ou en Java. Le chargement des données dans la base s'effectue sans traitement autre que le respect de la sémantique et du schéma global.

Le schéma d'intégration de BioWarehouse est défini de façon globale dans un fichier XML en deux parties. La première partie, appelée «CORE» définit l'ensemble des données, la seconde partie appelée «MAGE» est une extension pour gérer les annotations d'expressions géniques. Les tables du schéma relationnel sont définies à partir de schémas fréquemment rencontrés en biologie avec une unification des termes utilisés (utilisation d'ontologies) : ceci permet une intégration de données de sources diverses chargées à partir de différents modules.

L'implémentation de BioWarehouse a été prévue pour être utilisée selon un schéma relationnel et pouvant être utilisé avec des bases relationnelles libres comme MySQL ou commerciales comme ORACLE.

[68] http://biocyc.org/
[69] http://cmr.jcvi.org/tigr-scripts/CMR/CmrHomePage.cgi

### D) GenMapper

GenMapper[70] (Genetic Mapper) (Do and Rahm, 2004) intègre des données génomiques, biologiques et médicales provenant de 60 sources de données dont Entrez Gene, Unigene, UniProt, GO, InterPro, KEGG et OMIM.

L'une des caractéristiques de GenMapper est d'être basé non pas sur un schéma global (de type étoile ou flocon), mais sur un schéma générique, appelé GAM (Generic Annotation Management). Ce schéma permet une représentation uniforme de toutes les données intégrées dans l'entrepôt. En effet, le schéma repose sur deux classes principales que sont 'Source' et 'Objet', ce qui permet de représenter dans GAM chaque source comme associée à un ensemble d'objets (ou données contenues dans la source). Ainsi, le système est particulièrement bien adapté à l'ajout de nouvelles sources de données. Le réseau de cross-références existant entre les sources de données est exploité et contenu dans le schéma GAM.

GenMapper propose une interface conviviale de conception de requête, où l'utilisateur choisit son ou ses objets à analyser (par exemple, un ensemble de protéines). Il choisit ensuite les informations qu'il souhaite obtenir sur les objets de départ. Une vue sur GAM est générée et fournit à l'utilisateur une vision des données associées à ses objets de départ.

GenMapper n'intègre pas de données d'expression mais par ses capacités d'enrichissement de données, il est largement utilisé pour l'annotation et la recherche d'informations sur des groupes de gènes différentiellement exprimés.

### E) GEWARE

GeWare[71] (Gene Expression Warehouse) (Kirsten, et al., 2004) est un entrepôt de données qui intègre des données d'expression issues des puces à ADN Affymetrix, des informations sur les expériences et des données sur les gènes étudiés. Il supporte différents types d'analyses telles que le traitement des données d'expression, la visualisation de données, la création de groupes de gènes et l'analyse de ces groupes, des analyses OLAP.

Il est basé sur un modèle multidimensionnel relationnel où la table centrale de faits correspond aux données d'expression et où les dimensions correspondent aux annotations et aux traitements pouvant être effectués dans l'entrepôt. Les dimensions sont organisées en hiérarchies, les analyses OLAP permettent ainsi d'effectuer des opérations de drill-down* et de roll-up*, pour accéder à différents niveaux d'annotations.

GeWare fournit une interface Web servant pour l'intégration des données et les analyses. Le modèle générique GAM, décrit précédemment dans le système GenMapper,

[70] http://ducati.izbi.uni-leipzig.de:8080/GenMapper/servlet/gui.MainFrame
[71] http://ducati.izbi.uni-leipzig.de:8080/Geware/servlet/de.izbi.geware.common.forms.FrameSet

est utilisé pour capturer les annotations sur les gènes étudiés dans GeWare, les données sont ensuite transférées de GAM à la dimension concernée de GeWare.

# 4 DISCUSSION

Nous avons discuté dans ce deuxième chapitre les principales architectures issues de la recherche dans le domaine d'intégration de données, et qui sont soit des systèmes d'intégration matérialisée ou des systèmes d'intégration non matérialisée.

L'intégration réalisée par ces projets est soit horizontale, soit verticale, selon que les données considérées se complètent ou se chevauchent. Leur spécialisation respective les rend complémentaires, et aucun ne peut prétendre s'imposer comme la solution universelle au problème d'intégration de données biologiques. L'utilisateur doit donc faire son choix en fonction de la complexité du problème qu'il a à traiter.

L'approche matérialisée, ou entrepôt de données, telle que décrite en section 3.2, fournit deux avantages majeurs. Premièrement, le fait de stocker les données en local dans un schéma global facilite l'optimisation et l'exécution des requêtes. Deuxièmement, les données étant disponibles localement, l'approche permet aux utilisateurs d'ajouter leurs propres annotations, permettant ainsi de modifier, de valider et/ou de nettoyer les données intégrées. il est important de noter que l'entrepôt de données est la seule approche permettant de lutter efficacement contre les données inconsistantes provenant de différentes sources, mais également de fournir des moyens d'analyses avancés sur de grands volumes de données. Ainsi, même si la phase d'intégration est très couteuse lors de la conception d'un entrepôt de données, ceci est largement compensé par les capacités d'analyses ultérieures.

Les approches non matérialisées de type médiation ou navigationnelle sont des approches très récentes dans le domaine de la bioinformatique. Ce sont des approches conviviales et intuitives qui, contrairement à l'approche entrepôt de données, sont plutôt dédiées à des analyses ponctuelles, sur de faibles volumes de données. Leur avantage réside dans le fait d'interroger les sources en ligne et donc de disposer de données à jour. Cependant, les temps d'exécution sont très dépendants de la disponibilité et de l'accessibilité de ces sources externes.

La plupart des approches non matérialisées n'effectuent qu'une intégration horizontale des données en intégrant uniquement des sources de données complémentaires et rarement chevauchantes. En se limitant à des sources ayant des informations différentes sur des entités, on limite les capacités du système d'intégration en termes de fiabilité et de complétude. En effet, le système ne peut résoudre les problèmes liés aux données absentes ou contradictoires, ni identifier les données de mauvaise qualité. De même, le système ne

peut sélectionner les sources qui bénéficient de meilleurs temps de réponses aux requêtes et qui renvoient de meilleurs résultats sur les plans qualitatif et quantitatif. En plus, l'une des principaux inconvénients de l'approche de médiation est la difficulté de construction et de maintenance du schéma global sur lequel s'appuie le médiateur ; l'ajout ou le retrait d'une source oblige soit à le revoir entièrement (dans le cas de l'approche GAV), soit à ajouter un certain nombre de règles de correspondance (dans le cas de l'approche LAV), qui risquent de compliquer d'autant la phase de réécriture de requêtes.

De façon plus générale, les différents systèmes sont caractérisés par le langage ou le modèle de données dans lequel le schéma global est exprimé. Nous avons évalué les avantages et les inconvénients de l'utilisation de ces deux architectures pour les données biologiques et avons dressé un panorama des solutions existantes en informatique en montrant qu'elles ont été systématiquement appliquées aux données biologiques.

# Deuxième Partie

# CHAPITRE 3

# Utilisation d'une approche hybride pour l'intégration sémantique des données de Pseudomonas sp.

# Chapitre 3

# Utilisation d'une approche hybride pour l'intégration sémantique des données de Pseudomonas sp.

**Sommaire**



## 1 INTRODUCTION

Comme démontré en partie introductive de ce manuscrit, les données sont réparties sur le Web dans une multitude de sources de données dynamiques et très hétérogènes. Si depuis quelques années des efforts ont été fourni par la communauté scientifique pour améliorer l'interopérabilité entre ces différentes sources par la définition de standards et la proposition de différentes approches d'intégration, la problématique reste entière.

Au cours de ce travail de thèse, notre objective a été de fournir une solution d'intégration tenant compte des défis mentionnés ci-dessus et adaptée à notre contexte :

l'intégration de données biologique de *Pseudomonas sp*. Ce travail a été effectué dans le cadre d'un projet de collaboration entre le groupe **LABIPHABE** de la Faculté des sciences et techniques de Tanger et le groupe **Khaos** de l'école technique supérieure de l'ingénierie en informatique de l'université de malaga. Dans ce travail, nous avons visé à développer un entrepôt de données nommé ***PseudmonasDW***. C'est un entrepôt de données semi-structuré qui intègre des données enrichies à partir de sources génomiques, protéiques, métaboliques et enzymatiques. Les données sont nombreuse et de nature variées : il s'agit d'informations sur les séquences des gènes, leurs localisations chromosomiques, les protéines encodées, leurs implications dans des fonctions moléculaires et des processus biologiques, leurs implications cliniques, leurs niveaux d'expression dans différentes conditions physiopathologiques. Ajoutons à cela leur apparition croissante dans la littérature scientifique. Nous avons proposé une approche hybride qui vise à combiner les avantages des deux approches les plus connues dans le domaine d'intégration de données: (i) L'architecture entrepôt (approche matérialisée) qui est extrêmement bien adaptée à certains besoin du domaine biologique. L'utilisation d'un entrepôt est en effet souvent motivée par l'un au moins des trois points suivant. Premièrement, certains thèmes de recherche imposent une complète confidentialité des requêtes et un contrôle total des données où l'accès distribué est alors impossible. Deuxièmement, les recherches dans ce domaine font souvent appel à des traitements trop complexes pour tourner sur des données non rapatriées localement ou à des traitements nouveaux que l'on souhaite tester sur des données. Troisièmement, l'architecture entrepôt, lorsqu'une intégration sémantique est effectuée, permet de n'accéder qu'à des données nettoyées voire filtrés donc plus sûres et sur lesquelles on a une valeur ajoutée. (ii) Le système médiateur (approche virtuelle) qui est une approche duale dans laquelle les données restent stockées dans les sources. Le médiateur offre un accès transparent aux sources en donnant l'illusion qu'on interroge un système centralisé. Nous avons combiné les deux approches, virtuelle et matérialisée, pour exploiter leurs avantages dans un environnement hybride. D'une part l'entrepôt offre une bonne performance pour les données complexes et d'autre part, la mise à jour des données peut être réalisée, en cas de besoin, via le système médiateur.

La construction de ***PseudmonasDW*** s'est déroulé en plusieurs étapes y compris la définition des besoins, la conception du modèle de données, et enfin l'intégration des données.

**La définition des besoins** : cette étape est préalable à l'implantation de tout nouveau système d'information. L'étude des besoins nous a aidé à déterminer le contenu de ***PseudmonasDW*** et son organisation, ainsi que les requêtes que les utilisateurs formuleront. Cette étape est réalisée par le biais d'interviews auprès des futurs utilisateurs du système. Nous avons cherché à comprendre et à analyser les besoins qui pouvaient être exprimés par les biologistes lors du processus d'interrogation des sources de données publiques. Nous avons procédé de façon analogue à (Stevens, et al., 2001), qui propose une étude et une classification des tâches bioinformatiques effectuées dans l'analyse de données

génomiques, et qui recense les requêtes fréquemment posées dans l'analyse de données cliniques (Ely, et al., 2000). Plus particulièrement, nous avons cherché à mettre en évidence pourquoi une source de données était interrogée plutôt qu'une autre et comment les sources de données étaient interrogées. Les interviews nous ont permis de recenser les données à étudier et dans quelles dimensions. Ensuite, ces interviews nous ont aidé à identifier les sources requises pour l'intégration de données souhaitées.

**La conception du modèle de données** : L'ambition de ***PseudomonasDW*** est d'intégrer un ensemble de données provenant de sources variées, via un modèle global de données (voir section 2.1). La pertinence du système en termes de réponses aux requêtes reposes alors entièrement sur la pertinence de ce modèle. Pour réaliser notre modèle global de données ou le schéma intégrateur de l'entrepôt, nous avons agrégé les données provenant des différentes sources. Ainsi, des efforts ont été fournis pour :

- Respecter la fiabilité de l'information
- Respecter la cohérence des informations, une même données pouvant provenir de deux sources différentes, il faut alors choisir la plus judicieuse.
- Assurer la consolidation des informations, c'est-à-dire définir de manière unique une donnée.
- Unifier la représentation des données.
- Vérifier la non-redondance des informations.

**L'intégration des données :** c'est la procédure qui nous a permis de transformer les données des sources externes vers ***PseudmonasDW***, en les adaptant. En général, l'intégration de données au niveau d'un entrepôt est divisée en quatre étapes qui sont (i) l'extraction des données des sources. Cela consiste de collecter les données utiles des sources originales. (ii) La transformation des données aux niveaux syntaxique et sémantique. Cette étape permet de transformer, reformater et nettoyer les données afin d'éliminer les données non conforme au modèle de destination et d'éviter les doublons et autres incohérences. (iii) L'intégration des données et enfin (iv) le stockage local des données intégrées dans l'entrepôt. Il faut noter que cette décomposition est seulement logique. Dans ***PseudmonasDW***, l'étape d'extraction et une partie de l'étape de transformation ont été groupées dans le même composant logiciel appelé 'service de données' (ou service Web*). Une partie de l'étape de transformation et l'étape d'intégration ont été réalisées via le système médiateur SB-KOM (System Biology Khaos Ontology-based Mediator)(Navas-Delgado and Aldana-Montes, 2009). L'étape de stockage a été effectuée automatiquement en se basant sur quelques API (Application Programming Interface) de java.

# 2 VUE GLOBAL SUR LE SYSTEME PSEUDOMONASDW

Comme nous avons déjà décrit, ***PseudmonasDW*** (Pseudomonas Data Warehouse) est un entrepôt de données semi structuré qui permet l'intégration des données biologiques de l'espèce *Pseudomonas*. ***PseudomonasDW*** fournie des outils pour analyse des données intégrées, afin de mettre en évidence des corrélations entre les informations étudies. L'environnement regroupe au sein d'un seul et même modèle de données (schéma intégrateur) les instances provenant de ressources génomiques, protéiques, enzymatiques et métaboliques. Les instances du modèle sont ensuite interrogées par différentes APIs qui nous sommes antérieurement développées (voir section 3.2).

D'après Inmon, « L'entrepôt de données n'est pas un produit ou un logiciel mais un environnement. Il ne s'achet pas, il se bâtit » (Inmon, 2002). On distingue deux manières de construire un système d'intégration : *top-down* (Inmon, 2002), où l'on part de l'information souhaitée, pour ensuite chercher les sources pouvant répondre aux besoins, ou *bottom-up*, où l'on part de la volonté d'intégrer plusieurs sources de données (Kimball, 2003). Ainsi, dans les approches *top-down*, les schémas des sources importent peu pour la conception du schéma global. Ils seront seulement pris en compte dans un second temps quand les correspondances entre le schéma global et les schémas des sources seront établies pour permettre l'exécution de requêtes. Dans l'approche *bottom-up*, il faut noter que le schéma global fournisse une vue conciliée des différentes sources, impliquant une bonne connaissance au préalable des schémas des sources de données. Pour concevoir ***PseudmonasDW***, nous avons utilisé un processus d'intégration qualifié ascendant (*bottom-up*) où nous sommes d'abord partis du besoin de représenter au sein d'un même schéma telles et telles données, pour ensuite choisir les sources de données ainsi que les processus d'intégration appropriés. Par cette approche, nous relions de manière cohérente, les données génomiques avec les données enzymatiques et celles métaboliques, tout en assurant la réconciliation des données autour de la nomenclature des gènes. La combinaison des informations de plusieurs sources de données et des disciplines multiples permet une intégration forte et systématique, facilite la compréhension des processus cellulaire et par conséquence conduit à une prédiction des nouveaux comportements cellulaire.

## 2.1 Sources de données intégrées dans PseudmonasDW

Plusieurs sources de données pourraient être utilisées pour créer un entrepôt de données comme ***PseudmonasDW***. Dans la version actuelle, ***PseudmonasDW*** intègre cinq bases de données. Ces bases de données ont été sélectionnées pour leurs propriétés de contenu et de structuration les plus appropriés pour l'étude de *Pseudmonas sp ;* nous pouvons les

diviser en trois types : 1) bases de données génomique et protéique, 2) bases de données métabolique et 3) bases de données enzymatique. Une intégration forte des données du niveau génomique jusqu'à niveau métabolique, rend possible la réponse aux interrogations complexes posées par les chercheurs. Nous montrerons dans cette section pour chaque source de données, sa provenance, son contenu et sa structure.

### 2.1.1 Bases de données génomique et protéique

***PseudomonasDW*** offre une variété des données génomiques telle que l'annotation du gène et de protéine, gène de régulation, expression génique* (Gene expression) et une collection des facteurs de transcription. Ces données sont extraites à partir de trois bases de données :

- **GenBank** : c'est une base de données avec un accès libre. Elle est considérée comme une collection d'annotation pour toutes les séquences nucléiques qui sont publiquement disponible ainsi que leurs séquences peptidiques (Benson, et al., 2011). Cette base de données est produite au sein de NCBI (National Center for Biotechnology Information) comme une partie de la collaboration internationale des bases de données des séquences nucléotidiques (INSDC : Internatinal Nucleotide Sequence Database Collaboration). GenBank et ses collaborateurs reçoivent les séquences produites dans les laboratoires de recherche pour plus de 380 000 organismes. Elle est accessible via le système de NCBI Entrez qui intègre des données de grandes bases de données de séquences d'ADN et de protéines avec la taxonomie, le génome, le mappage, la structure et les domaines d'information de la protéine, et la littérature via le journal biomédical PubMed. GenBank est une des premières banques de données qui ont proposé le format XML pour présenter leurs enregistrements avec une DTD bien définie pour spécifier la structure et la terminologie du domaine pour leurs enregistrements des gènes et des séquences soumises.

- **Uniprot** : (base de données universelle de protéines) est la plus grande des bases de données informatique pour les protéines de tous les organismes vivants et les virus (Consortium, 2010). Elle fournit des informations sur la fonction des protéines, leur structure ainsi que des liens vers d'autres bases de données. Elle combine les données de Swiss-Prot, TrEMBL et Protein Information Resource (PIR) et elle est met à jour régulièrement. Ses données reposent sur le serveur ExPASy[72] de l'Institut suisse de bioinformatique. Uniprot contient 534242 séquences entières contenant 189454791 acides aminés extraites de 206707 références[73]. Uniprot offre les données en format HTML, XML et Fasta.

[72] http://expasy.org/

[73] Release 2012_01 of 25-Jan-12 >> http://web.expasy.org/docs/relnotes/relstat.html

- **PRODORIC**[74] : est un acronyme de **PRO**cariotIC **D**atabase **O**f Gene-**R**egulation. Cette base de données est basée sur une approche intégrée. elle fournit des informations sur les réseaux moléculaires chez les procaryotes avec un accent sur les organismes pathogène (Münch, et al., 2003). Actuellement PRODORIC contient principalement des informations détaillées sur les structures des opérons* et des promoteurs, y compris une énorme collection des sites de liaisons et de facteurs de transcription. Aussi qu'un nombre approprié des sites de liaison régulateurs est disponible et une matrice du poids de position (position weight matrix) est fourni. Ces données sont recueillies manuellement par le dépistage de la littérature scientifique originale. PRODORIC offre un service web pour accéder à plusieurs parties de la base de données. Les utilisateurs peuvent accéder à l'API du serveur du PRODORIC par la technologie SOAP via le protocole HTTP en utilisant un langage informatique spécifique de leur choix. Le serveur SOAP fournit également un fichier WSDL (Web Service Description Language. Cela permet aux utilisateurs d'intégrer dynamiquement des requêtes de PRODORIC dans leurs propres programmes

### 2.1.2 Bases de données métaboliques

**KEGG** est une encyclopédie des gènes et des génomes, elle a été lancée par le programme humain japonais de génome en 1995 (Minoru, 1997). Selon ses réalisateurs, KEGG est considérée comme étant une « représentation d'ordinateur » du système biologique (Kanehisa, et al.). KEGG relie les informations connues au-dessus des réseaux moléculaires, comme les voies et les complexes (c'est la base de données des voies), les informations sur des gènes et protéines produit par des projets de génome (base de données des gènes) et les informations sur les composés biochimiques et les réactions (bases de données des réactions). Ces bases de données sont des différents réseaux connus respectivement sous les noms de réseau de pathways, l'univers de gènes et l'univers chimique.

Dans notre cas, nous nous sommes intéressés que par la base de données des voies (KEGG PATHWAY) qui offre des voies métaboliques et quelques autre processus cellulaires. Nous avons accédé au serveur API du KEGG par le biais de la technologie du SOAP via le protocole HTTP. Le serveur SOAP est accompagné d'un fichier WSDL qui facilite la construction d'une bibliothèque client pour un langage informatique spécifique. Cela nous a permis d'écrire notre propre programme et d'automatiser la procédure d'accession au serveur API du KEGG et finalement d'obtenir les résultats souhaités (Kanehisa, et al.).

[74] http://www.prodoric.de/

### 2.1.3 Bases de données Enzymatique

***PseudomonasDW*** offre des données enzymatiques extraites de la base de données enzymatique **BRENDA** (Chang, et al., 2009). Cette base de données représente la collection principale des informations concernant la fonctionnalité des enzymes disponibles à la communié scientifique. Elle est disponible gratuitement via internet et aussi comme une base de données interne pour les utilisateurs commerciaux. BRENDA est maintenue et développée à l'institut de biochimie et de bioinformatique au sein de l'université technique de Braunschweing, en Allemagne. Les données sur la fonction enzymatique sont extraites directement de la littérature primaire par des scientifiques titulaires d'un diplôme en biologie ou en chimie. Les vérifications formelles et de cohérence sont effectuées par des programmes informatiques, chaque ensemble de données sur une enzyme classée est vérifiée manuellement par au moins un biologiste et un chimiste.

Le contenu de BRENDA couvre des informations sur la fonction, la structure, l'occurrence, la préparation et l'application d'enzymes. Les outils d'analyse et de gestion des données ont été mises en œuvre pour améliorer le traitement, la présentation, la saisie et l'accès aux données. BRENDA offre désormais de nouvelles options d'affichage telles que l'affichage des paramètres fonctionnels, la vue 3D de la séquence de protéines et des caractéristiques de la structure.

## 2.2 Architecture de l'intégration des données biologiques au sein de PseudmonasDW

D'une communauté à l'autre, l'entrepôt est une architecture dans laquelle les données sont plus ou moins structurées ainsi que plus ou moins historisées. On trouve dans la littérature(Calvanese, et al., 1998) la distinction de deux approches dans la construction d'entrepôts respectivement appelées approches *procédurale* et *déclarative*.

- Dans l'approche procédurale les données sont intégrées de façon ad-hoc sans chercher à construire un schéma intégrateur. Dans le cas où aucune structure ni aucun historique ne sont imposées aux données, on parlera plus souvent de la notion de dépôt de données (*ou data repository*) que d'entrepôt de données (ou *data warehouse*).

- Dans l'approche déclarative (Calvanese, et al., 1998) la structuration des données de l'entrepôt se fait grâce à son schéma global, ou schéma intégrateur. Le modèle dans lequel le schéma global est défini détermine le langage de requêtes utilisé pour interroger l'entrepôt.

Pour ***PseudomonasDW***, nous avons choisi l'approche déclarative qui malgré sa complexité reste majoritairement suivie. L'approche déclarative nous a motivé à réaliser notre contribution en faisant appel au système médiateur et l'architecture entrepôt pour une intégration hybride et forte au sein d'un schéma global. Ce schéma regroupe les instances provenant des diverses sources intégrées et nous a garanti un échange de données d'une façon compréhensible. Le système médiateur que nous avons utilisé, *SB-KOM* (System Biolgy Ontology-based Mediator)(Navas-Delgado and Aldana-Montes, 2009), est basé sur une infrastructure nommée *KOMF* (Chniber and Kerzazi, 2008). Le *KOMF* est une infrastructure générique pour enregistrer et gérer les ontologies, leurs relations et les informations reliées aux ressources. Cette infrastructure est basée sur un middleware nommé '*SD-Core*' (Navas-Delgado and Aldana-Montes, 2009). Une description détaillée de cette infrastructure est présentée dans la section 3. *KOMF* a été instancié avec succès dans le contexte de la biologie moléculaire pour l'intégration des sources de données biologiques qui sont accessible via le web (Briache, et al., 2012).

Dans cette section, nous décrivons l'architecture générale du notre entrepôt de données. ***PseudmonasDW*** est composé de plusieurs composants indépendamment implémentés et jouent des rôles différents et complémentaires dans le processus de l'intégration de données. La Figure 18 montre une représentation schématique de l'architecture du système.

La couche de sources représente la base du système et elle constitue le point d'accès aux bases des données *KEGG* (Kanehisa, et al., 2006), *BRENDA* (Chang, et al., 2009), *Uniprot* (Consortium, 2010), *GenBank* (Benson, et al., 2011) et *PRODORIC* (Münch, et al., 2003).

Derrière le système entrepôt de données se place toute la logistique pour établir un flux de données entre ***PseudmonasDW*** et les bases de données intégrées. Cela s'est achevé via le processus ETL (Extract-Transform-Load) (Thomas and Stefan, 2008). Il s'agit d'une technologie informatique intergicielle (comprendre middleware) permettant d'effectuer des synchronisations massives d'information d'une base de données vers une autre. Ce processus repose sur des connecteurs servant à exporter ou importer les données dans les applications, des transformateurs qui manipulent les données, et des mises en correspondance (mappages). Notre objective de l'utilisation du processus ETL est l'intégration et la réexportation de données des sources originales dans ***PseudmonasDW***.

Dans le système ***PseudmonasDW***, les bases de données publiques sont uniformément accédées et interrogées par le médiateur *SB-KOM* (System Biology Khaos Ontology-based Mediator) (Navas-Delgado and Aldana-Montes, 2009). Le médiateur offre des interfaces d'adaptateurs pour les sources de données et aussi transforme les données dans un modèle de données commun utilisé par *SB-KOM*. Le système ***PseudmonasDW*** est constitué d'un ensemble des services de données (un service de données pour chaque source de données) qui encapsulent la fonctionnalité des adaptateurs. Ces derniers

occupent une partie très importante dans les éléments internes des services de données. Un adaptateur reçoit une requêtes XQuery à partir du *SB-KOM*, la transforme en une requête appropriée à la source de données qui le convient, performe tous les traitements supplémentaires et retourne un document XML au médiateur. Le rôle du service de données est de permettre à l'administrateur de ***PseudmonasDW*** d'utiliser les fonctionnalités des adaptateurs pour interroger et extraire les informations sollicitées à partir des sources de données via leurs pages web ou le mécanisme FTP.

Le *SB-KOM* utilise les ontologies comme des schémas intégrateurs dans le but de performer la réécriture des requêtes et par conséquence l'activation de la fonctionnalité de l'étape de transformation. Autrement dit, les réponses des requêtes XQuery – matérialisées au niveau des documents XML - sont envoyées à *SB-KOM* qui les transforme et les combine en une instance du schéma intégrateur (ou schéma global). Les résultats finaux obtenus sont donc chargés au niveau de l'entrepôt de données et fournis aux utilisateurs au format HTML.

Dans ce contexte, le processus ***ETL*** (Extract-Transform-Load) s'initialise par l'intervention de l'administrateur du ***PseudmonasDW***. Ce dernier choisit l'information qu'il souhaite extraire puis sélectionne l'espèce à stocker dans l'entrepôt de données. Ensuite, le système extrait automatiquement toutes les données souhaitées par le biais des services web. Finalement, le système transforme les données extraites en un format commun en utilisant les différents composants de *SB-KOM*. Notre proposition est d'utiliser une ontologie pour l'intégration de données, où chaque source de données est reliée avec le schéma global par des règles de correspondances définies (mappings).

Le stockage de données dans ***PseudmonasDW*** se fait d'une manière intergicielle en utilisant quelques bibliothèques de Java (Exemple : Jena[75] et Java DOM[76]). Nous avons aussi utilisés eXist[77] qui nous a permis de stocker automatiquement nos données dans un entrepôt de données XML natif. Une description détaillée de différents composants du système est cité dans la section suivante.

[75] http://jena.apache.org/
[76] http://docs.oracle.com/javase/1.4.2/docs/api/org/w3c/dom/package-summary.html
[77] http://exist.sourceforge.net

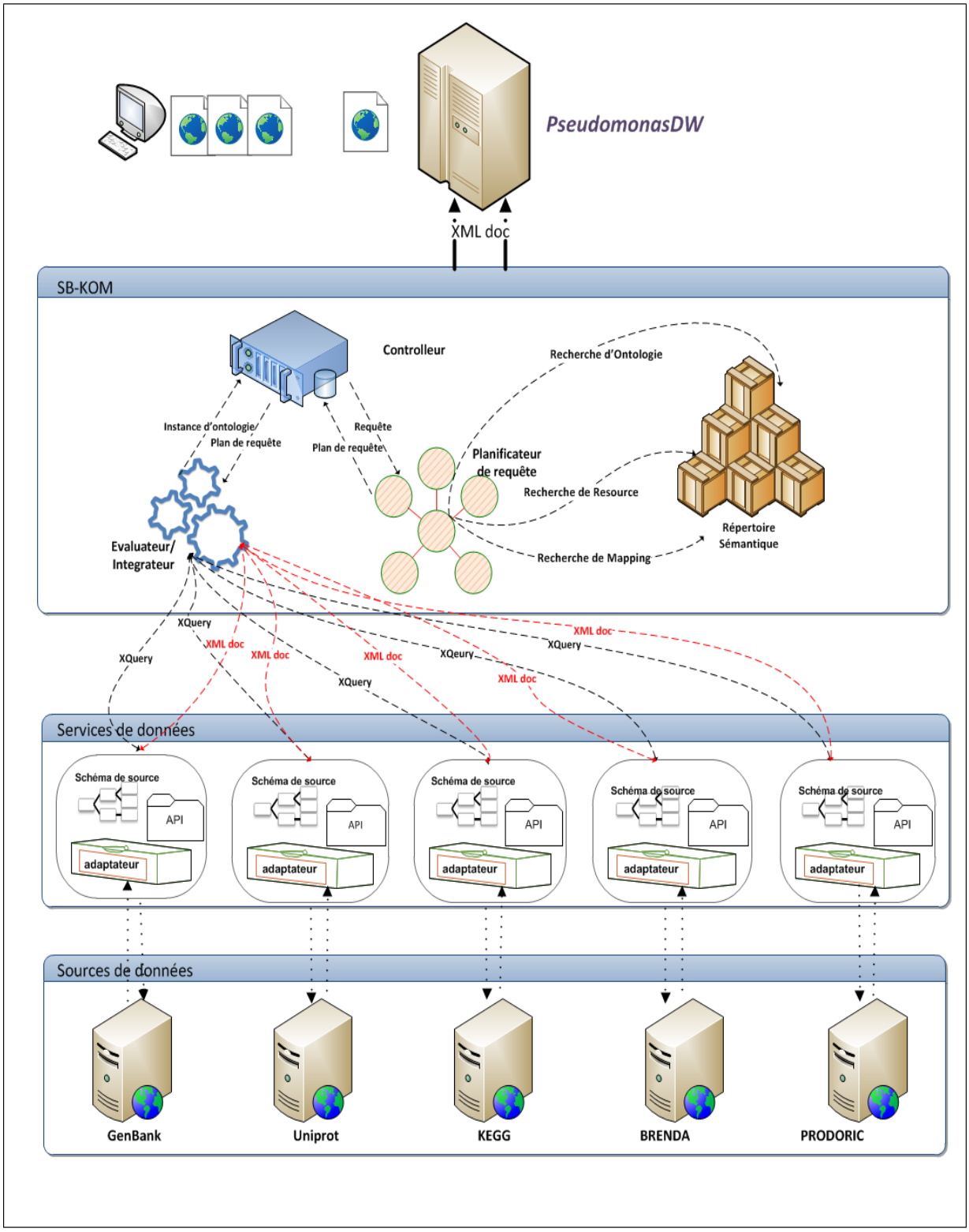


**Figure 18. Les différentes couches constituant le système PseudomonasDW**

# 3 DIFFERENTS MODULE D'INTEGRATION AU SEIN DE L'ENTREPOT DE DONNEES PSEUDOMONASDW

Comme nous avons déjà mentionné dans les paragraphes précédents, nos objectifs dans cette thèse sont (i) l'inclusion de données génomiques de haut débit (ii) l'intégration de plusieurs sources de données en utilisant une approche hybride permettant l'utilisation d'un système médiateur pour une intégration sémantique au sein d'un entrepôt de données. (iii) le maintien de données de ***PseudmonasDW*** à jours avec celles des bases de données d'origine.

En générale, l'intégration de données dans ***PseudomonasDW*** a été effectuée selon deux niveaux : le premier niveau est l'intégration syntaxique qui consiste à extraire les données de sources originales et les transformer en un modèle uniforme (XML) utilisé par *SB-KOM*. Nous avons choisi XML –autrement dit XML, XML schema et XQuery- comme un modèle de données commun. Le deuxième niveau d'intégration est appelé intégration sémantique qui consiste à convertir les données extraites en terme du schéma global du ***PseudomonasDW*** en créant des règles de correspondance entre chaque schéma de source et celui de l'entrepôt. ***PseudomonasDW*** a un ensemble de modules qui dépend fortement à des technologies de XML et de web sémantique. Dans ce qui suit, nous donnons une description détaillée sur les différents composants de ***PseudomonasDW***.

## 3.1 Schémas de source

La modélisation des connaissances du domaine d'application de ***PseudomonasDW*** constitue la pierre angulaire pour l'intégration efficace de données. Pour cela, une étude détaillée des sources a été effectuée dans le but d'établir une terminologie standard pour décrire les données. Chaque source de données a été modélisée par un schéma exporté.

Un schéma est un ensemble d'éléments connectés par une certaine structure. En pratique, il existe différentes représentations, qui sont le modèle relationnel, le modèle orienté objet ou le XML. Dans chacune des représentations, on distingue des éléments et des structures : les entités et les relations dans le modèle relationnel, les objets et les relations dans le modèle orienté objet et les éléments et les sous-éléments dans le XML.

Comme une première étape dans la construction de ***PseudmonasDW***, nous avons créé un schéma XML pour chaque source de données (Figure 19). Ces schémas sont considérés comme des modèles qui décrivent les données et leur organisation dans les sources de données. Ils définissent la structure sous laquelle les résultats seront retournés

de services de données. Les schémas de sources nous ont permis d'avoir une idée globale sur les différentes données qui seront représentées sur le schéma intégrateur de l'entrepôt.

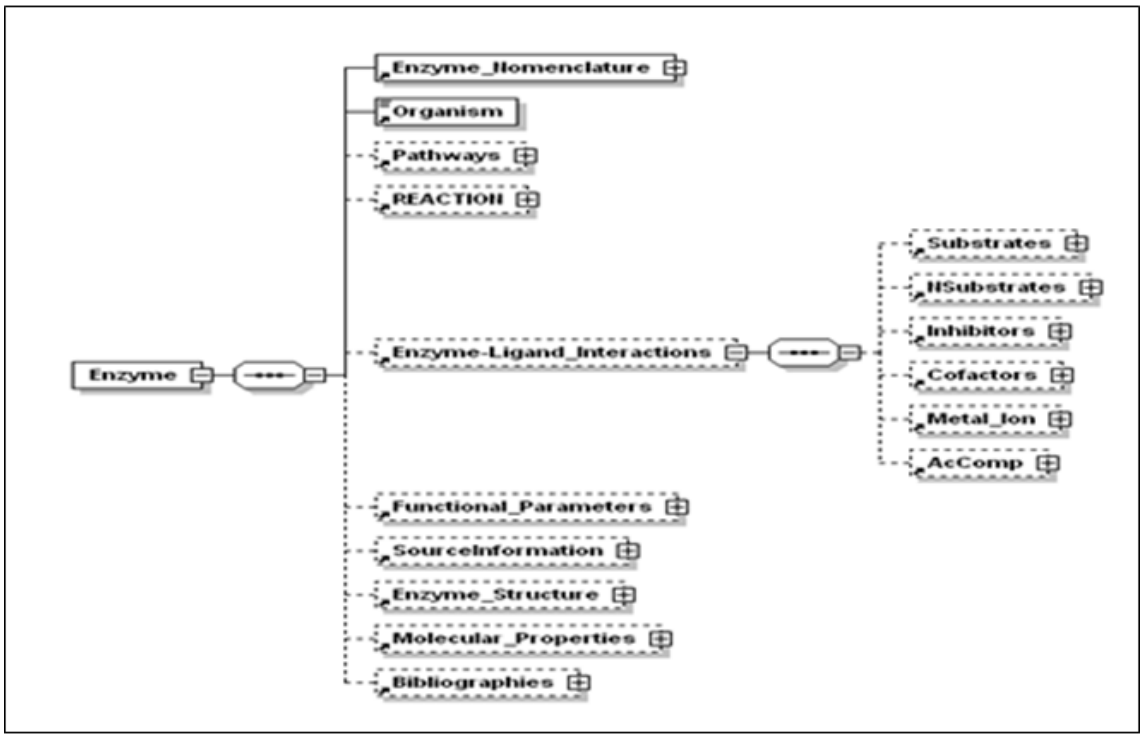


**Figure 19. Un fragument représentatif du schéma XML de la source de données BRENDA**

## 3.2 Services de données

Il est bien connu qu'un adaptateur est une interface pour interroger les sources de données et transformer les données en un modèle de données utilisé par le système d'intégration (Levy, 1999). Puisque le but de ***PseudomonasDW*** est d'intégrer des bases de données accessibles via le protocole web, il est complètement normal qu'un adaptateur est considéré comme le composant le plus important dans l'architecture du système. Nous avons développé cinq adaptateurs sémantiques chacun pour une base de données. Nous pouvons définir l'adaptateur sémantique comme un adaptateur qui peut gérer les connaissances du Web.

Nous avons proposé d'améliorer le processus de l'implémentation des adaptateurs en les publiant comme des services Web (service de données dans notre cas) qui peuvent être réutilisés par autres systèmes d'intégrations. Les services Web permettent l'invocation de fonctions distantes, présentes sur des systèmes distribués et hétérogènes, grâce au protocole HTTP et à XML. Selon (Kadima and Monfor, 2003), « les services Web sont des

applications auto-descriptives, modulaires et faiblement couplées qui fournissent un modèle de programmation et de déploiement d'applications, basé sur des normes, et s'exécutent au travers de l'infrastructure Web ». Et selon (Zimmermann, et al., 2006) « un service est un composant applicatif mis à la disposition sur un réseau et disposant de méthodes que l'on peut invoquer à distance via l'emploi de protocoles standard. Les services Web présentent l'avantage d'être faiblement couplés, indépendants des plateformes et réutilisables »

Le but des services de données est de permettre à ***PsudomonasDW*** d'accéder à la fonctionnalité des adaptateurs. Dans ce contexte, nous avons conçu une architecture adaptative avec laquelle nous avons pu définir un service de données comme «un service Web qui offre des fonctionnalités d'interrogation par les adaptateurs en utilisant le protocole Web ».

### 3.2.1 Architecture du service de données dans PseudmonasDW

Dans cette section, nous présentons notre architecture du service de données (Figure 20). Elle inclut un ensemble d'outils qui nous a aidé à extraire les données de *Pseudomonas sp* de différentes sources de données.

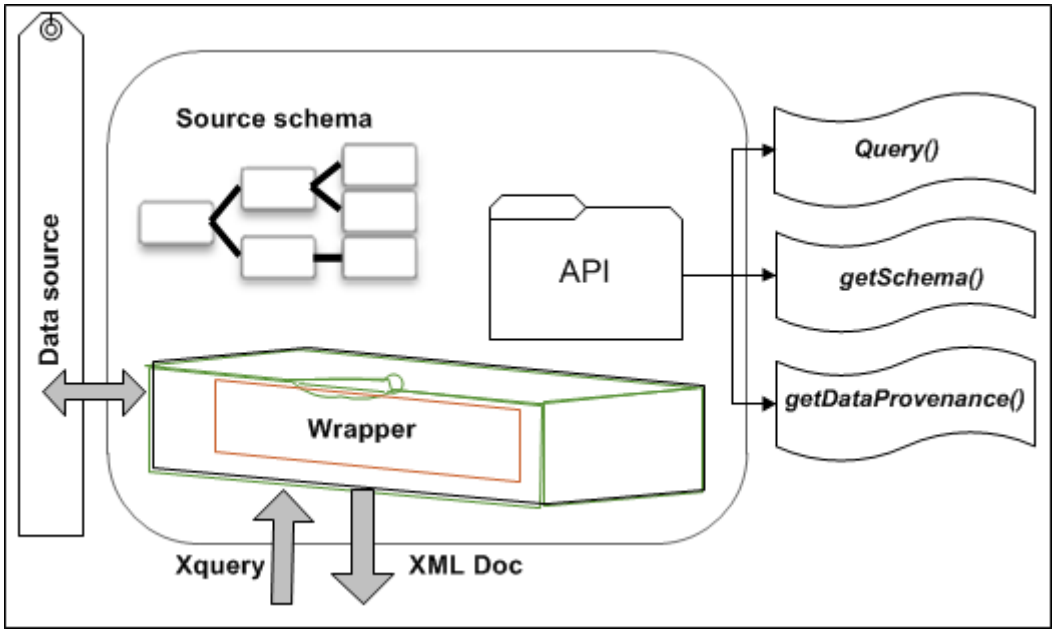


**Figure 20. Représentation schématique de l'architecture du service de données dans le système PseudmonesDW**

Ce type de service utilise un processus bidimensionnel : (1) pour accéder aux sources de données en utilisant l'adaptateur qui traite une requête et retourne un document

XML ; (2) pour l'exportation de fonctionnalités d'interrogations par l'adaptateur et sa sémantique comme un service web. La sémantique du service Web inclut des informations sur le schéma de la source et la provenance de données. Cette dernière est nécessaire, dans le domaine de la bioinformatique, dont il est très important de savoir quelle source de données a été utilisée dans l'extraction d'une telle donnée. Dans ce contexte, en plus de service de requête de l'adaptateur, le service de données enveloppe une API (Application Programming Interface).

L'API constitue le point d'accès à la fonctionnalité du service Web. Elle publie trois méthodes : *Query()* qui soumit la requête XQuery à l'adaptateur et retourne un document XML. La structure du ce document doit satisfait les contraintes du schéma de la source. Les deux autres méthodes, *getschema()* et *getDataprovenance()*, permissent l'accès aux métadonnées stockées dans le service Web. La méthode *getschema()* retourne le schéma XML de la source de données et la méthode *getDataprovenance()* fournit des informations sur la base de données interrogées (par exemple le nom de la base de données).

Derrière le service Web, il y a une spéciale classe java qui traite l'appelle aux différentes méthodes. Cette classe s'appelle la classe *Service* ; qui est un composant générique conçu pour définir les trois différentes méthodes qui reçoivent l'appelle au service Web. La partie importante de la classe Service est de tenir la correspondance entre la requête XQuery (Hunter, 2003) et le langage de requête sous-jacent de la source de données. Autrement dit, la classe *service* est responsable de mettre des correspondances entre les paramètres de la requête XQuery et les paramètres de la source de données.

### 3.2.2 Implémentation du service de données dans PseudmonasDW

Pour publier nos services de données comme des services Web, nous avons utilisé Apache Tomcat[78] comme un serveur d'application et Axis[79] comme une plateforme pour présenter le Web service. La première étape dans la publication du service web était la copie de tous les fichiers des classes java qui nous avons programmé, les bibliothèques utilisées et le fichier descripteur de déploiement dans le répertoire WEB-INF du répertoire racine du service de données (Figure 21). Le descripteur de déploiement est un fichier nommé *web.xml* qui contient tous les caractéristiques et les paramètres du web service.

[78] http://tomcat.apache.org/
[79] http://ws.apache.org/axis/overview.html

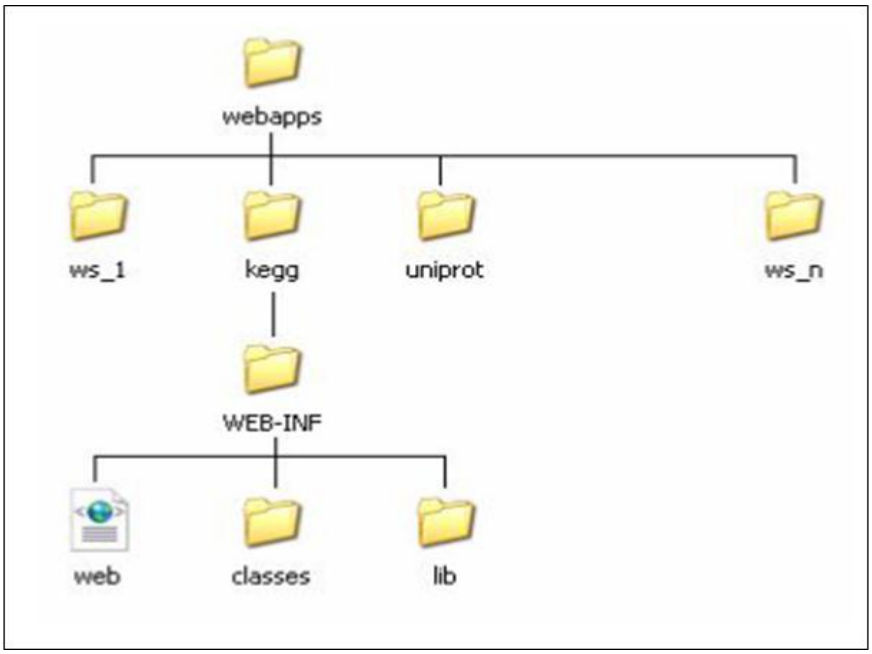


**Figure 21. Première étape de déploiment du service Web**

La deuxième étape du déploiement du service web était la création du fichier *deploy.wsdd* dans le même dossier que le *web.xml*. Ce fichier contient l'ensemble des propriétés de déploiement du notre service Web qui ont été exprimées par l'élément <service> (Figure 22).

```
<deployment xmlns="http://xml.apache.org/axis/wsdd/"
            xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
    <service name="webServiceName" provider="java:RPC">
        <parameter name="className" value="package.Service"/>
        <parameter name="allowedMethods" value="*"/>
    </service>
</deployment>
```

**Figure 22. Deuxième étape de déploiement du service Web**

Les attributs de l'élément <service> définissent les caractéristiques principales du service Web dont:

- L'attribut *name* indique le nom du service web.
- L'attribut *provider* définit le type de fournisseur de service qui était utilisé pour réaliser l'implémentation du service Web. Nous avons utilisé le *provider*

Java RPC qui permet d'exposer une classe Java quelconque en tant que service Web.

Le restant des propriétés du service Web a été défini par le biais d'éléments <parameter> qui définissent le nom et la valeur de différentes propriétés :

- Le paramètre *className* a été utilisé pour spécifier le nom complet de la classe d'implémentation Java du service. La valeur de ce paramètre est le chemin vers la classe java compilée associée au service Web (nous referons ici à la classe *Service*).
- Le paramètre *allowedMethod* a été utilisé pour définir la liste des méthodes exposées par le service Web. La valeur spéciale * indique que nous avons exposés toutes les méthodes du serveur Web.

La dernière étape de déploiement du service Web était la déclaration du service dans le fichier de configuration du serveur. Pour cela nous avons utilisé l'outil d'administration d'Axis *AdminClient* auquel nous avons fournis en paramètre le descripteur de déploiement du service via la commande suivante :

```
java -classpath %AXISCLASSPATH%
        org.apache.axis.client.AdminClient deploy.wsdd
-http://hostname:portnumber/webServiceFolderName/services/AdminService
```

Cette opération nous a permis de mettre à jours le fichier Tomcat/webapps/Service Web/WEB-INF/*server-config.wsdd*. La vérification du bon déploiement du service Web a été effectuée par la saisie de la direction 'http://hostname:portnumber/webserviceName/Services' dans la barre d'adresse du navigateur. Cela nous a permis d'obtenir les déférentes méthodes définies dans le service Web (Figure 23).


**Figure 23. Capture d'écran de differentes méthodes du service Web àprès déploiement**

## 3.3 Schéma Intégrateur du PseudmonasDW

Comme nous avons mentionné avant, ***PseudomonasDW*** vise à intégrer un ensemble de sources de données biologiques hétérogènes dans un seul système. Dans l'approche déclarative (Calvanese, et al., 1998), suivie dans ce travail, la structuration des données de l'entrepôt se fait grâce au schéma global. Le schéma intégrateur (global) peut intégrer les données à différents niveaux. Nous pouvons distinguer **l'intégration syntaxique** qui a été effectuée par les services de données et consiste à convertir l'ensemble des données des sources dans le modèle choisi pour l'entrepôt. À cette étape, le schéma global de l'entrepôt est constitué de l'union des schémas des sources. Si les sources offrent chacune des informations sur des entités différentes, cette intégration est suffisante pour n'avoir aucune redondance au niveau du schéma intégrateur.

Néanmoins, ***PseudomonasDW*** intègre des sources de données offrant des informations chevauchantes. Une agrégation d'information a été alors requise pour identifier des objets équivalents d'un point de vue sémantique, c'est-à-dire nous avons appliqué une **intégration sémantique** pour supprimer toute redondance au niveau du schéma de l'entrepôt. L'intégration sémantique est fondée sur la construction d'un schéma global intégrateur et vise à convertir les données des sources en termes des données dans ce schéma global intégrateur.

« Le schéma global correspond à la description des relations entre toutes les données partagées dans le système sans aucune description de leur implémentation ou de leur stockage physique, il garantit un échange de données d'une façon compréhensible » (King, et al., 2008).

En général, la mise en œuvre d'un système intégrateur de données exige la détermination de la manière par laquelle le schéma global sera spécifié (par exemple : quel modèle de données doit être adopté et quel type de contraintes sur les données peut être exprimé). Pour ***PseudmonasDW***, nous avons suivi l'approche GAV (Global-As View) qui consiste à définir le schéma global en fonction des schémas locaux des sources de données (voir chapitre 2). Notre propose est d'utiliser une ontologie (***PseudomonasDW Ontology***) comme un schéma global de l'entrepôt. Notre ontologie a été construite par la réconciliation de tous les différents schémas de sources en une seule ontologie cohérente (Figure 24).

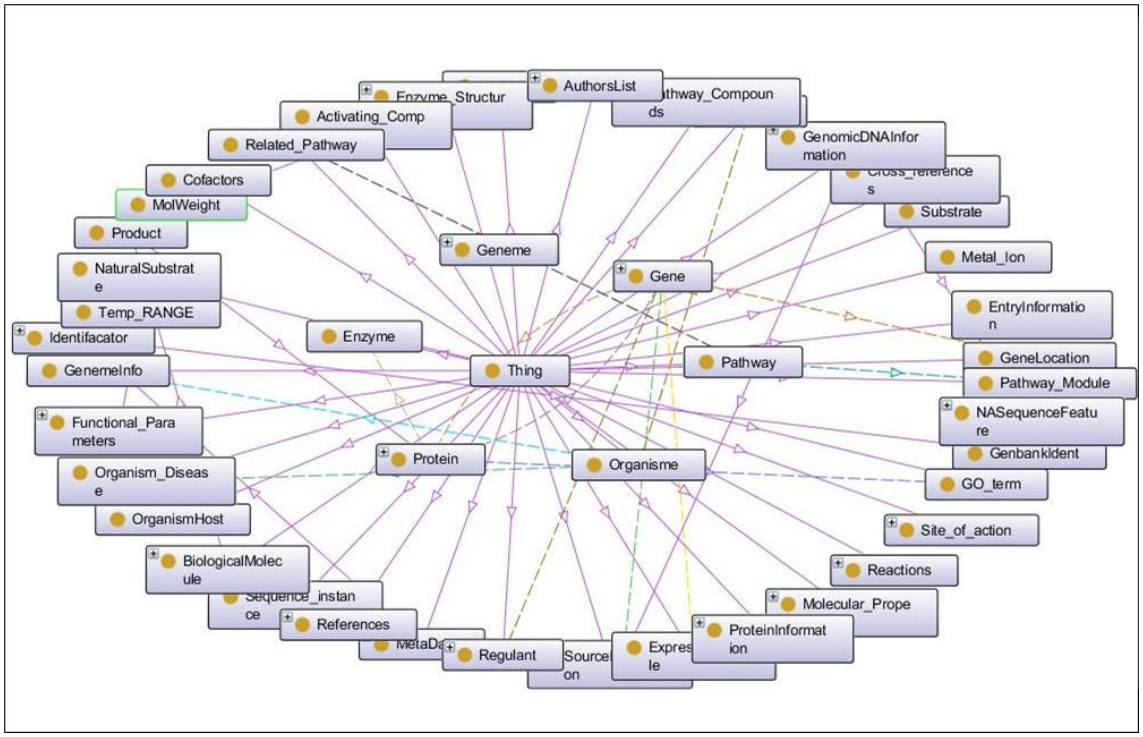


**Figure 24. Quelques conceptes de l'ontologie de domaine de PseudomonasDW**

Dans le contexte du Web sémantique, l'ontologie de domaine est utilisée comme un schéma pour l'intégration de données. Le principe d'un tel schéma est de fournir une interface unique pour l'interrogation de sources de données hétérogènes. Pratiquement, une ontologie de domaine est plus générale et sémantiquement plus riche qu'un simple schéma conceptuel.

Une ontologie de domaine est une « description intentionnelle de ce qui nous connaissons autour de l'essence des entités d'un domaine particulier en utilisant des concepts et des relations entre ces concepts» (Sun and Liu, 2006). L'ontologie de domaine de ***PseudomonasDW*** organise, sous forme d'une hiérarchie, les connaissances sur notre domaine en regroupant les entités du domaine en sous catégories suivant ses caractéristiques. Notre ontologie de domaine est principalement utilisée comme une terminologie pour la description explicite et cohérente de nos données. Elle assure l'encapsulation sémantique des sources de données en définissant la hiérarchie de concepts. Elle est considérée comme une classification de toutes les entités biologiques manipulées par l'entrepôt. L'ontologie de ***PseudmonasDW*** représente un modèle de connaissance qui modélise des connaissances biologiques et bioinformatique dans un cadre conceptuel simple limité par des relations parent-enfant de type '**isA**'. L'enfant est une classe qui représente un sous-ensemble des éléments du parent ; chaque enfant hérite toutes les propriétés de son parent en plus des siennes spécifiques. Les concepts de l'ontologie

peuvent être classés en deux catégories : la catégorie des concepts biologiques et la catégorie des concepts reliés aux sources de données.

- Les concepts biologiques représentent toutes les classes qui modélisent les entités biologiques. (par exemple les classes : gene, genome, protein, enzyme…)
- Les concepts reliés aux sources de données sont représentés par des classes référant directement aux sources de données. Nous citons comme exemple le concept *Source* qui représente les sources biologique intégrées dans l'entrepôt et le concept *Entry* qui représente les entrées dans les sources de données originales. Ce type de concept a un rôle très important pour garder les traces de données dans ***PseudmonasDW***.

Pour des informations sémantiques additionnelles, l'ontologie définie deux types de propriétés: (i) propriétés des objets (*object properties*) qui représentent les relations entre les individus d'une ou deux classes différentes. (ii) propriétés des types de données (*datatype properties*) qui relient un individu avec des types de données. L'ontologie de ***PseudmonasDW*** contient 110 classes, 79 propriétés des types de données et 44 propriétés des objets.

Pour mieux illustrer le rôle des propriétés dans la transmission de la sémantique au niveau de l'ontologie, nous détaillons un exemple du monde réel (Figure 25) dont les éclipses représentent les concepts, les flèches continues représentent les propriétés des objets alors que les flèches discontinues représentent les propriétés des types de données. Le gène ***algU*** code pour la protéine '*RNA polymerase sigma-H factor*' qui est un facteur d'initiation qui promeuve l'attachement de l'ARN polymérase à des sites d'initiation spécifiques (Martin, et al., 1993). Ce facteur sigma régule des gènes comme ***algD*** (code pour la protéine '*GDP-mannose 6-dehydrogenase*') qui est impliqué dans la synthèse d'alginate (Roychoudhury, et al., 1992).

- Les deux gènes ***algU*** et ***algD*** codent respectivement au régulateur '*RNA polymerase sigma-H factor*' et l'enzyme '*GDP-mannose 6-dehydrogenase*'.
- ***algU*** régule le gène ***algD***.
- Les gènes ***algU*** et ***algD*** codent pour des protéines ayant respectivement les mêmes abréviations que leurs gènes.
- Le régulateur a le nom '*Sigma-30'* comme un nom alternatif.
- L'enzyme à un numéro de classification enzymatique qui égale à 1.1.1.132.

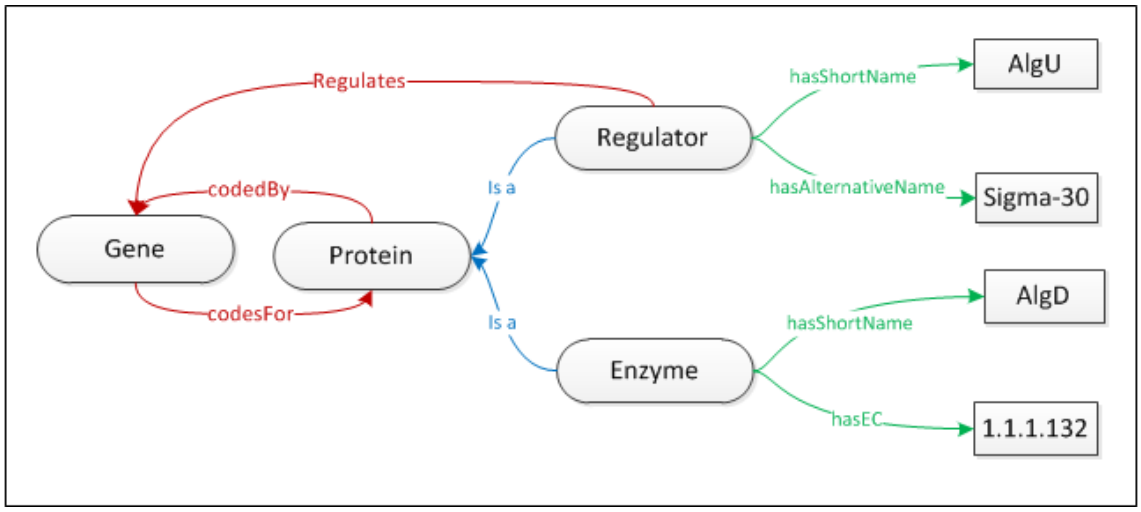


**Figure 25. Représentation schématique de l'exemple traité dans cette section. Il montre quatre conceptes biologiques (éclipses) liées par des propriétés d'objet (dlèches rouges), deux relation parent-enfant (flèches bleues) et deux propriétés de données (flèches vertes).**

A partir de cet exemple nous pouvons déduire :

- Quatre concepts : '*Gene*', '*Protein*', '*Regulator*' et '*Enzyme*'.
- Trois propriétés d'objets : '*codefor*' et son inverse '*codedBy*' qui relient les deux concepts '*Gene*' et '*Protein*' plus la propriété '*Regulates*' qui relie '*Regulator*' au '*Gene*'.
- Trois propriétés des types de données : '*hasShortName*' pour les deux concepts '*Regulator*' et '*Enzyme*', '*hasAlternativeName*' pour le concept '*Regulator*' et enfin '*hasEc*' pour le concept '*Enzyme*'.
- Les deux concepts '*Regulator*' et '*Enzyme*' sont considérés comme des enfants du concept '*Protein*'.

Dans ***PseudmonasDW***, nous avons choisi OWL comme un langage d'ontologie standard. Pour être plus précis, nous avons utilisé OWL-Lite (qui un sous langage de OWL) parce que nous avons envisagé dès le départ de développer une simple ontologie de domaine qui présente une simple hiérarchie des concepts.

## 3.4 Correspondances sémantiques entre les schémas

En plus de la modélisation de l'ontologie et des schémas de sources, nous avons eu besoin d'établir des associations entre les différents concepts de l'ontologie et les éléments appropriés qui représentent l'information dans les sources de données. L'établissement de ces correspondances sémantiques est une tâche difficile. Elle constitue actuellement une

des étapes les plus coûteuses lors du développement d'un système d'intégration de données (Toumani, et al., 2007).

Comme nous avons déjà cité, nous avons utilisé l'approche GAV (Global-As View), qui exige que le schéma global de l'entrepôt doive être exprimé en termes des sources de données. Cela signifie que chaque concept et propriété de l'ontologie représente une vue définie en termes de différents éléments des sources de données. Cette vue détermine la manière d'obtenir des instances du schéma intégrateur à partir des sources de données.

Les associations entre les concepts de l'ontologie et les éléments des schémas de sources (Figure 26) sont matérialisées au sein de ***PseudomonasDW*** par des règles de correspondance (mappings). Ces règles sont utilisées pour permettre la transmission de données en termes de l'ontologie de système. Dans ce contexte, les règles de mappings que nous avons utilisées sont définies comme un pair $(P,Q)$, dont :

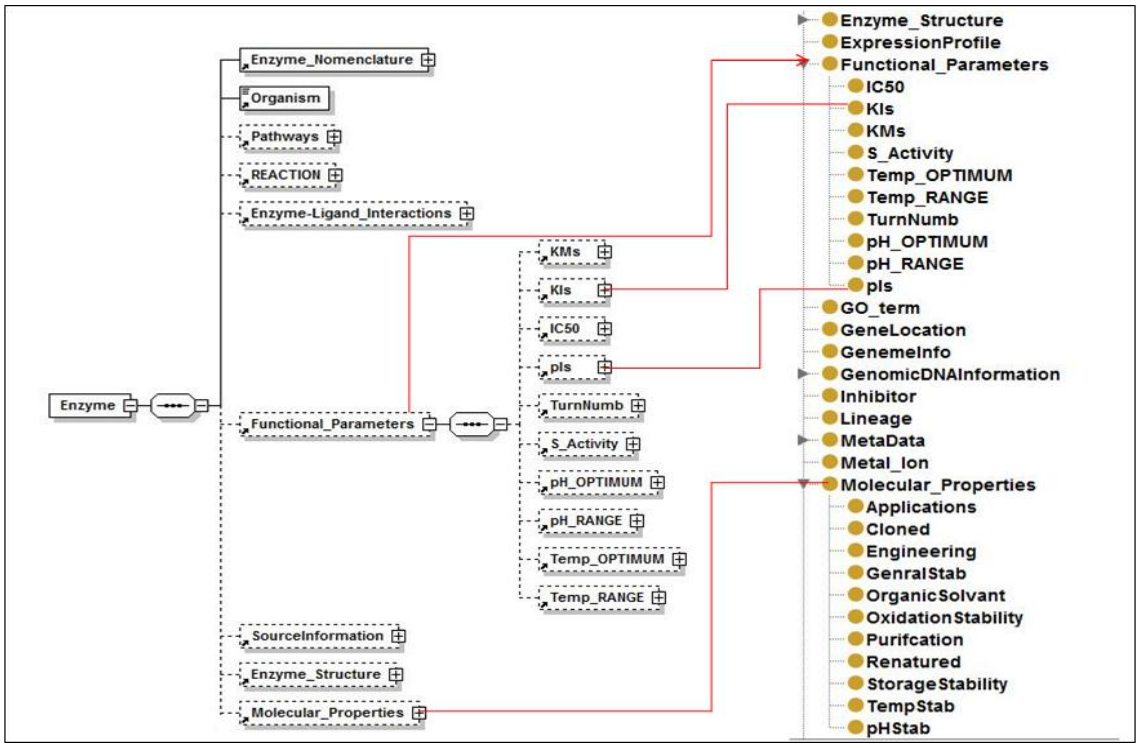


**Figure 26. Associations entre les concepts de l'ontologie de domaine de PseudomonasDW et les éléments d'une partie du schéma XML de la source de données BRENDA**

- $P$ est une ou un couple d'expressions du chemin exprimées en XPath.
- $Q$ est une requête conjonctive exprimée en termes des concepts de l'ontologie.

En générale nous avons définie trois types de mappings :

**Mapping des Classes** : ce type de mappings définie des associations entre les classes de l'ontologie et les schémas de sources. Ce type de mapping s'écrit de la manière suivante :

*XPath-Element-Location, Ontology-Class-Name, correspondence-index*

Le fragment '*XPath-Element-Location*' représente la position d'un élément du schéma d'une source exprimée en XPath. Le fragment '*Ontology-Class-Name*' représente le nom de la classe correspondante au niveau de l'ontologie. La partie '*correspondence-index*' est un indice représenté par un nombre entier qui détermine la justesse de l'instance du mapping. Dans ***PseudomonasDW***, cet indice égale toujours à 100 puisque toutes les associations sont faites manuellement. Ci-dissus un exemple de mapping qui associe les classes '*Enzyme*' et '*KM*' avec leurs correspondants dans le schéma du BRENDA.

```
/Result/Enzyme,Enzyme,100
```

```
/Result/Enzyme/Functional_Parameter/KM,KM,100
```

**Mapping des propriétés de type de données**: ce type de mapping associe les propriétés de type de données au niveau de l'ontologie avec les schémas de sources. Il s'écrie comme suit :

*XPath-Domain-Location; XPath-value-Location, Ontology-Domain-Name; Property-Name, correspondence-index*

Le fragment '*XPath-Domain-Location*' décrit le chemin vers un élément du schéma qui est associé avec le domaine de la propriété de type de données. Le fragment '*XPath-value-Location*' représente l'élément dont la propriété a eu la valeur de son rang. Les deux fragments '*Ontology-Domain-Name*' et '*Property-Name*' représentent respectivement le domaine et la valeur de la propriété. L'exemple suivant concerne la propriété de type de données '*hasValue*'

```
/Result/Enzyme/Functional_Parameter/KM;/Result/Enzyme/Functional
_Parameter/KM/KM_Value,KM;hasValue,100
```

```
/Result/Enzyme/Functional_Parameter/pH_Optimum;/Result/Enzyme/Fu
nctional_Parameter/pH_Optimum/pH_Optimum_Value,pH_OPTIMUM;hasVal
ue,100
```

**Mapping des proprieties d'objets**: ce type de mapping associe les propriétés d'objets au niveau de l'ontologie avec les schémas de sources. Il s'écrie de la manière suivante :

*XPath-Domain-Location; XPath-Range-Location, Ontology-Domain-Name; Ontology-Range-Name; Property-Name, correspondence-index*

Les deux fragments '*XPath-Domain-Location*' et '*XPath-Range-Location*' décrivent les chemins des deux éléments qui correspondent au domaine et le rang de la propriété d'objet au niveau du schéma. Les deux fragments '*Ontology-Domain-Name*' et '*Ontology-Range-Name*' représentent respectivement le domaine et le rang au niveau de l'ontologie. Le fragment '*Property-Name*' correspond au nom de la propriété d'objet. L'exemple suivant montre comment la propriété d'objet '*hasFunctionalParameter*' est associée au schéma de source.

```
/Result/Enzyme;/Result/Enzyme/Functional_Parameter,Enzyme;Functi
onal_Parameter;hasFunctionalParameter,100
```

## 3.5 SD-Core: Genetic Semantic Middleware Components for the Semantic Web

Le rôle essentiel d'un middleware est de gérer la complexité et l'hétérogénéité des infrastructures distribuées. D'une part, le middleware offre des abstractions de programmation qui cachent certains des complexités du développement d'une application distribuée. D'autre part, une infrastructure d'un logiciel complexe est nécessaire pour mettre en œuvre ces abstractions. Autrement dit : au lieu qu'un programmeur doive traiter tous les aspects d'une application distribuée, le middleware peut s'occuper de certains d'entre eux.

Dans ce contexte, nous avons utilisé un middleware, précédemment développé par le groupe khaos (Navas-Delgado, 2008) pour profiter de ses composants dans l'intégration de données de *Pseudomonas sp* dans notre entrepôt. L'infrastructure de ce middleware est basée sur un répertoire de ressource 'resource directory', nommé SD-Core (Semantic Directory Core). le groupe Khaos a défini le SD-Core comme « un ensemble d'éléments de base pour construire des applications de Web sémantique, il est disponible en tant que serveur pour enregistrer la sémantique fournie par les services d'interrogations et aussi pour consulter toutes les sémantiques enregistrées » (Navas-Delgado and Aldana-Montes, 2008). L'utilisation de SD-Core nous a offert la moyenne de l'interopérabilité sémantique avec le médiateur SB-KOM. Dans le but de bien définir les éléments internes du répertoire sémantique (Semantic Directory).

Ainsi, le SD-Core est composé de deux ontologies inter-reliées OMV (Hartmann, et al., 2005) et SDMO, qui décrivent les sémantiques internes du répertoire sémantique(Figure 27). OMV enregistre des informations additionnelles sur les ontologies alors que SDMO est l'ontologie qui se charge de l'enregistrement des informations sur les ressources, les relations entre ces ressources ainsi que les ontologies enregistrées dans OMV.

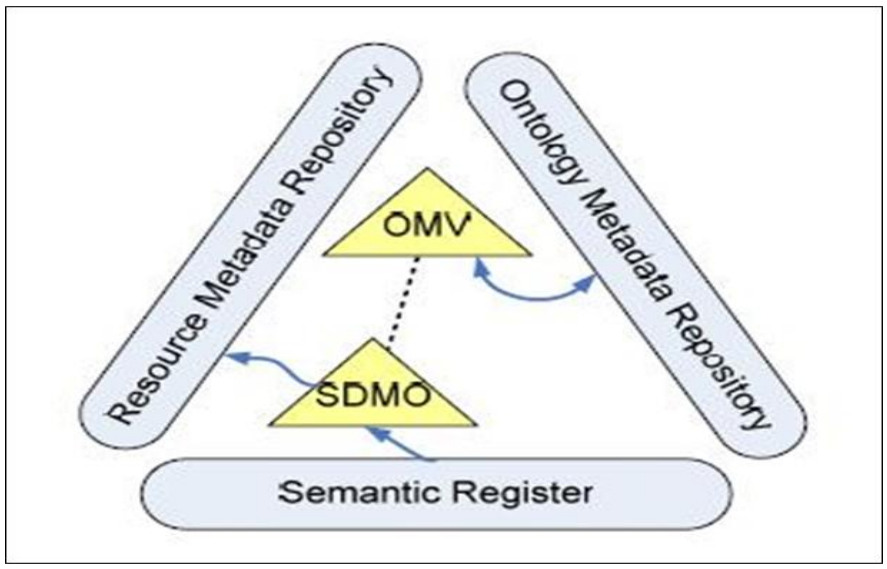


**Figure 27. Les différentes interfaces et ontologies constutiant le SD-Core**

Le SD-Core est composé de trois interfaces qui regroupent un ensemble minimum des éléments pour construire un grand nombre d'applications pour le Web Sémantique.

*L'interface de répertoire des métadonnées de l'ontologie*: est une interface qui offre différents types d'accès aux informations reliées aux ontologies enregistrées au niveau de SD-Core. Les méthodes suivantes représentent quelques-unes de celles fournies par le middleware pour enregistrer et consulter les ontologies : *registerOntology(url,name), getOntology(name), getOntology( url), listOntologies()* and *listOntologies(concept)*.

*L'interface du registre sémantique*: se charge par les ressources relatives aux ontologies enregistrées au niveau du SD-Core. Lors de l'enregistrement des ressources, les implémentations de l'interface génèrent une instance de SDMO qui contient les correspondances (mappings) entre le schéma de cette ressource et les ontologies enregistrées au niveau du SD-Core. Cette interface offre des méthodes qui permettent aux utilisateurs d'enregistrer des ressources ainsi que ses mappings (exemple *registerResource( serviceName, url, queryMethod, schemaMethod)*).

*L'interface du répertoire des métadonnées de la ressource* est considérée comme une interface d'accès aux informations des ressources via des méthodes définies.

Le SD-Core offre une interface web (Figure 28) qui nous a permis d'accéder aux différentes fonctionnalités du Middleware et d'enregistrer notre ontologie de domaine, nos services de données ainsi que les schémas de sources et les mappings. Cette étape nous a permis d'enregistrer notre sémantique et toutes les informations nécessaires pour les rendre disponibles pour le médiateur SB-KOM dans le but de parser, écrire, planifier, optimiser et

solutionner les requêtes provenantes de l'administrateur du ***PseudmonasDW*** (plus de détail dans la section 3.6).

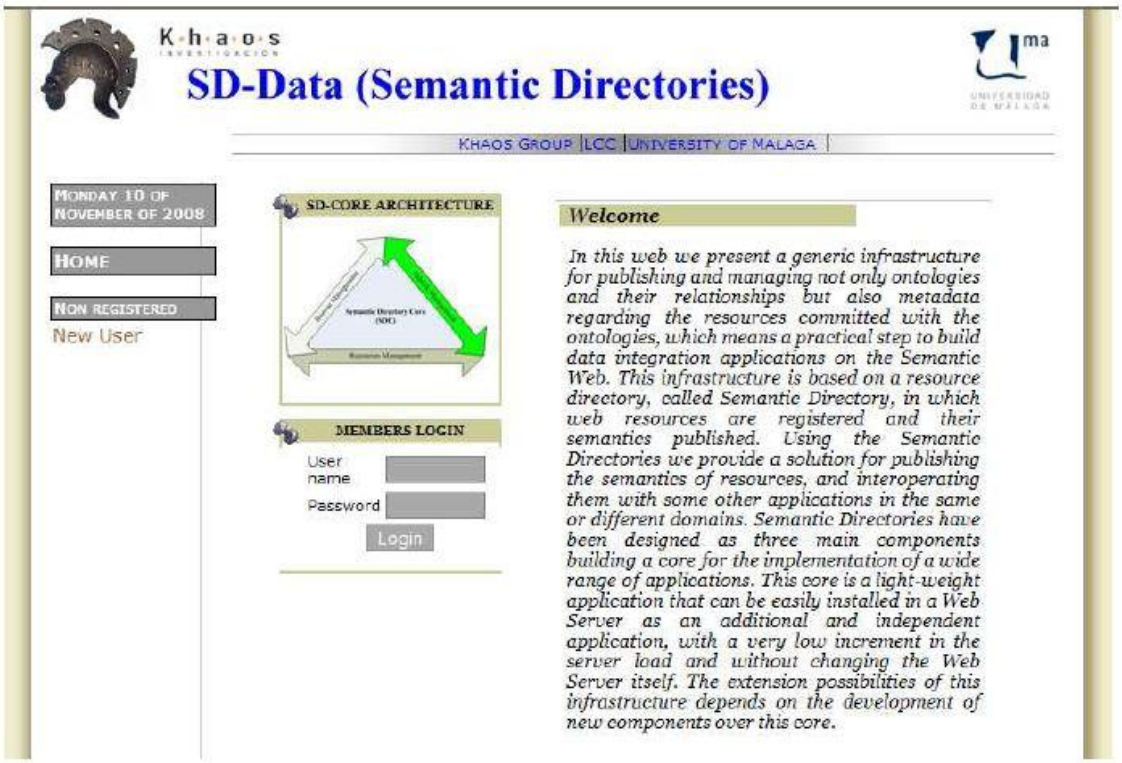


**Figure 28. L'interface Web SD-Core qui permet l'accès aux fonctionalités du Middelware et l'enregistrement de la sémantique nécessaires pour le médiateur SB-KOM**

## 3.6 SB-KOM: System Biology Khaos Ontology-based Mediator

Pour intégrer les données de *Pseudomonas* dans notre entrepôt, nous avons visé à utiliser un système médiateur précédemment développé par le groupe khaos. Ce médiateur nommé SB-KOM (System Biolgy Ontology-based Mediator) (Navas-Delgado and Aldana-Montes, 2009) qui est basé sur le KOMF (Chniber and Kerzazi, 2008). KOMF est une infrastructure générique pour enregistrer et gérer les ontologies, leurs relations et les informations reliée aux ressources. Cette infrastructure est basée sur le SD-Core. KOMF a été instancié avec succès dans le contexte de la biologie moléculaire pour l'intégration des sources de données biologiques qui sont accessible via le web. Le médiateur SB-KOM est composé de trois principaux composants : *le contrôleur, le planificateur de requêtes et l'évaluateur/intégrateur*.

*Le contrôleur* reçoit des requêtes du l'administrateur du ***PseudmonasDW*** et coordonne les autres composants du médiateur pour évaluer ces requêtes et obtenir des

résultats. Le contrôleur crée des fils pour les différentes requêtes de ***PseudmonasDW*** et assume le rôle d'un middleware entre les autres composants du SB-KOM. Les requêtes provenant de l'administrateur de l'entrepôt sont exprimées comme des prédicats conjonctifs (Hillebrand, et al., 1995), avec trois types principaux de prédicat : les classes en terme de l'ontologie de domaine enregistrée au niveau de SD-Core, les propriétés de type de données qui relient les individus aux valeurs latérales et les propriétés d'objets qui relient les individus entre eux. Les résultats de ces requêtes sont des instances de l'ontologie de domaine.

*Le planificateur de requêtes* est un des pillées les plus importantes pour l'élaboration des plans de requêtes pour traiter les requêtes soumises par ***PseudmonasDW***. Les plans générés par ce composant déterminent quelles sources de données doivent être utilisées pour extraire les informations souhaitées et dans quel ordre doivent être interrogées.

Selon la requête conjonctive soumise par l'administrateur de ***PseudomonasDW***, il y aura différents types de mappings au niveau du SD-Core. Les classes de l'ontologie de domaine de ***PseudmonasDW*** seront connectées à XPath d'un ou plusieurs éléments des schémas XML des sources de données. D'autre part, les propriétés de types de données seront connectées à deux expressions : la première correspond à la classe et la deuxième correspond à la propriété. Les propriétés d'objet seront liées aux classes dont leurs XPath sont actives dans la propriété.

Le *planificateur de requêtes* s'exécute selon un algorithme simple qui reçoit une requête conjonctive exprimée en termes de l'ontologie de ***PseudmonasDW*** (une conjonction de concepts et de propriétés) et retourne un ensemble des plans possibles sous forme d'arbres. Les étapes de l'algorithme sont énumérées en-dessous :

1. Distribuer tous les prédicats de la requête (concepts et propriétés) en deux groupes en se basant sur le nombre d'arguments : $G_1$ contient les prédicats ayant un argument (les concepts) et $G_2$ contient les prédicats ayant deux arguments (les propriétés)
2. Construire $G_S$: un ensemble de combinaisons entre les deux groupes en se basant sur le nombre d'arguments, ajouter tous les éléments de $G_1$ et $G_2$ à cet ensemble et éliminer les éléments répétés.
3. Eliminer les éléments de $G_S$ qui n'ont pas une représentation dans les mapping enregistrées au niveau de SD-Core.
4. Elaborer un plan sous forme d'arbre pour chaque variable instancié dans les arguments prédicats :
    a. La variable instanciée constitue le nœud racine.
    b. Les éléments qui contiennent un prédicat spécifiant une valeur pour la variable instanciée et les éléments qui ne contiennent que la variable instanciée (sans les autres variables) seront passés au nœud courant et éliminés de $G_S$.

c. Les éléments qui contiennent une autre variable en plus de celle instanciée, constitueront les arcs entre le nœud actuel et d'autres nouveaux et seront éliminés de $G_S$. Les nouveaux nœuds crées seront représentés par d'autres variables qui seront des variables instanciées.
d. S'il y a encore des éléments dans $G_S$, continuer dans l'étape 4.b pour chaque nouvelle variable instanciée.

*L'évaluateur/Intégrateur* est le troisième composant du SB-KOM. il analyse le plan de requête (QP) et performe des appels correspondantes aux services de données impliqués dans les sous requêtes (SQ1,…,SQn) du plan QP. Pour répondre à la requête de l'administrateur de ***PseudomonasDW***, ce composant exécute les services de données dans l'ordre spécifié au niveau du plan QP. Ensuite, les adaptateurs extraient les données souhaitées de sources originales et retournent des documents XML. L'intégrateur construit des instances (des modèles RDF) à partir des résultats des services de données en utilisant les mappings. Ces instance ne sont pas connectées entre elles parce qu'elles proviennent de services de données différents. Afin d'obtenir des instances associées, l'intégrateur établie des relations entre elles en utilisant les propriétés d'objets définis dans l'ontologie de domaine et qui sont représentées comme des relations entre les services dans le plan de requête. Finalement, ces instances associées sont filtrées afin d'éliminer les informations inutiles.

# 4 PROCESSUS ETL DANS PSEUDOMONASDW

Dans cette section nous traitons un exemple avec lequel nous essayons d'expliquer comment interviennent les différents composants de ***PseudmonasDW*** dans le processus d'ETL (Extraction, Transformation and loading). Cet exemple traite une requête soumise par l'administrateur de l'entrepôt. Nous prenons comme exemple la requête conjonctive suivante envoyée par l'administrateur de l'entrepôt :

```
"Ans(P,E,O,G,PW):-
Protein(P),hasPrteinName(P,"ProteinName"),ForOrganism(P,O),Enzym
e(E),IsEnzyme(P,E),Organism(O),hasOrganismName(O,"OrganismName")
,ForOrganism(E,O),Gene(G),CodedBy(P,G),PathWay(PW),ParticipateIn
(P,PW);"
```

Cette requête a pour but de chercher des informations sur une protéine nommée : '*ProteinName*' (exemple : *Acetyl-coenzyme A carboxylase carboxyl transferase subunit alpha*) pour un organisme appelé '*OrganismName*' (*Pseudomonas fluorescens (strain Pf-5))*. Avec la soumission de cette requête, l'administrateur cherche des informations concernant la protéine, les voies

métaboliques dans lesquelles intervient cette protéine, l'enzyme qui la corresponde et des données sur le gène qui code pour elle.

Cette requête conjonctive inclue trois types de prédicats principaux : Classes en terme de l'ontologie de ***PseudomonasDW*** exemple de *Protein(P)*, des propriétés de type de données qui relie les individus avec des valeurs latérales exemple de *hasProteinName (P, "Value")* qui relie la protéine avec son nom, et finalement les propriétés d'objet qui relient les individus entre eux comme *isEneyme(P,E)*. En général, cette requête est composée de cinq classes (*Protein, Organism, Enzyme, Gene* et *Pathway*), deux propriétés de types de données (*hasproteinName* et *hasOrganismName*) et quatre propriétés d'objects (*ForOrganism, IsEnzyme, CodedBy* et *ParticipateIn*) (Figure 29).

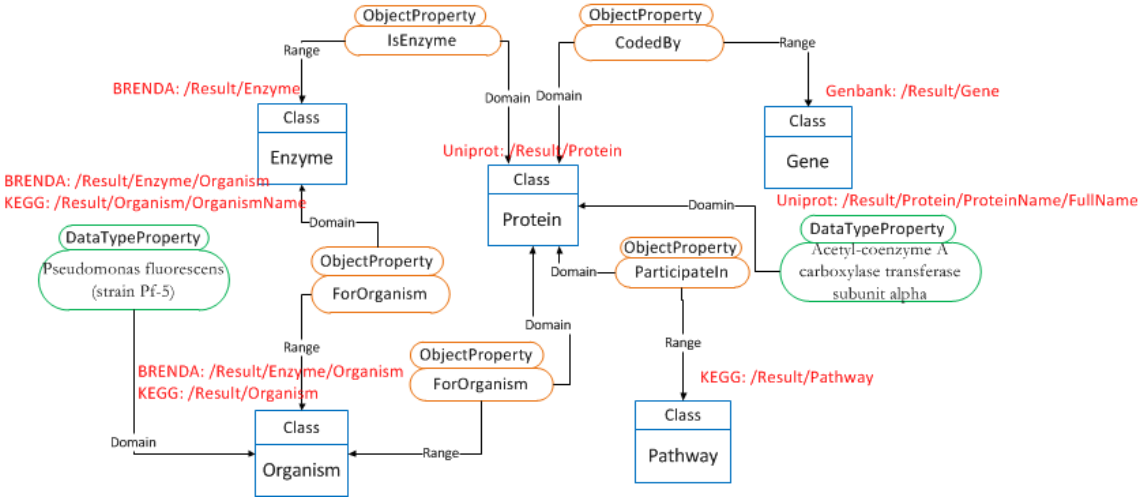


**Figure 29. Un schéma représentatif du fragment de l'ontologie qui intervient dans la formulation de la requête XQuery. les classes sont représentées en bleu, les propriétés d'objet sont représentées en orange et les propriétés de données sont représentées en vert. les règles de correspondances entre les schémas des sources et l'ontologie de domaine sont écrites en haut des éléments de l'ontologie en rouge**

La requête retourne les instances de la classe protéine qui a le nom *"ProteinName"* et qui sont reliées aux :

- *"Organism"* par le biais de la relation *"ForOrganism"*
- *"Pathway"* par la relation *"ParticipateIn"*
- *"Enzyme"* par le biais de la relation *"IsEnzyme"*. Cette enzyme est reliée aussi à la classe *"Organism"* par la relation *"ForOrganism"*
- *"Gene"* par la relation *"CodedBy"*.

Comme une étape antérieure, la requête conjonctive est envoyée au SB-KOM. Une fois la requête est reçue au niveau du contrôleur, une demande sera envoyée au planificateur de la requête. Ce composant utilise son algorithme basé sur les prédicats de la

requête et les règles de correspondance enregistrées au niveau du répertoire sémantique 'SD-Core'. Cet algorithme va générer un ensemble de sous-requêtes et aussi un plan d'exécution. Les prédicats de la requête conjonctive sont divisés en deux types : un ensemble qui contient les prédicats ayant un seul argument et un autre qui contient les prédicats ayant plus qu'un argument. Les prédicats qui ont des arguments communs et appartiennent aux deux ensembles sont ensuite regroupés dans des groupes représentés par la combinaison de deux ou plusieurs prédicats. Les groupes qui ne sont pas représentés par le mapping enregistré au niveau du SD-Core sont éliminés. Toutes les sous-requêtes possibles générées par le contrôleur sont représentées dans **la Table 3**.

A partir de cet ensemble de sous-requêtes, le planificateur va essayer de construire des arbres potentiels de l'ordre d'exécution. Il sélectionne les groupes qui ont des variables instanciées pour définir la racine de l'arbre. L'ordre de l'exécution du plan dépend aux variables instanciées : les groupes ayant des variables instanciées sont les premiers à exécuter, ensuite les groupes qui sont reliés à ces variables, et ainsi de suite jusqu'à l'exécutions de tous les groupes. Dans notre cas, G1 et G7 sont sélectionnés : G7 ne peut pas jouer le rôle d'un nœud racine parce qu'il n'y a aucun group qui lui dépend. Contrairement à G1 qui peut servir comme racine et par conséquent sera le premier groupe à exécuter (Figure 30). G1, et à près son exécution, renvoie des informations relatives à la protéine (P) du G8. Ensuite, G2, G3, G4 et G5 sont exécutés en parallèle parce qu'ils dépendent aux variables instanciées de G1. A partir de ses exécutions simultanées, l'algorithme va déterminer tous les objets reliés à la protéine (P) par les relations *"ForOrganism"*, *"CodedBy"*, *"ParticipateIn"* et *"IsEnzyme"*. Une fois ces objets sont obtenus, l'algorithme va exécuter les groupes G9, G10, G11 et G12. Puisque le groupe G6 dépend au groupe G12, ils seront exécuté à la fois pour obtenir des instances de l'Enzyme (E).

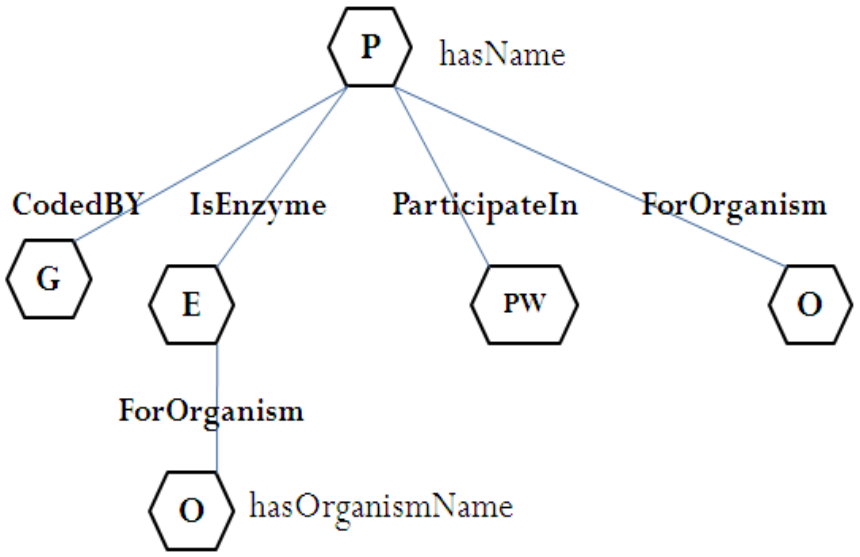


**Figure 30. Le plan de requête du l'exemple précédemment décrit. Chaque noeud et arc contient des informations pour accéder aux services de données**

**Table3 : Les différents groups intervenant dans la construction du plan de requête.**

| Groupe | Sous-requête | Service de Données |
|---|---|---|
| G1 | Protein (P), hasName | Uniprot |
| G2 | ForOrganism (P,O) | KEGG |
| G3 | CodedBy (P,G) | Genbank |
| G4 | ParticipateIn (P,PW) | KEGG |
| G5 | IsEnzyme (P,E) | BRENDA |
| G6 | ForOrganism (E,O) | BRENDA |
| G7 | Organism (O), hasOrganismName | Uniprot |
| G8 | Protein (P) | Uniprot |
| G9 | Organism (O) | BRENDA, KEGG |
| G10 | Gene (G) | Genbank |
| G11 | Pathway (PW) | KEGG |
| G12 | Enzyme (E) | BRENDA |

Les arcs de l'arbre de planification sont représentés par les propriétés d'objets, alors que les nœuds représentent les concepts de l'ontologie (Figure 30). Chaque arc et chaque nœud contiennent toutes les informations nécessaires pour l'exécution des sous-requêtes par le composant évaluateur/l'intégrateur. Ces informations se composent de : la sous-requête (élaborée à partir du mapping) exprimée en XQuery et correspond au nœud ou à l'arc du plan, le nom et la direction du service de données à exécuter.

Les services de données de ***PseudmonasDW*** sont exécutés par le composant Evaluateur/l'intégrateur en suivant le plan d'exécution généré par le planificateur. Pour notre cas, le service de données de 'Uniprot' reçoit la première sous-requête parce que la propriété de type de données hasProteinName est mappé au schéma XML de Uniprot. Le nom du gène codant pour '*Acetyl-coenzyme A carboxylase transferase subunit alpha*', le numéro de classification d'enzyme (Ec number) relatif à la protéine, les noms des vois métaboliques dans lesquelles elle participe sont obtenus comme une réponse de la sous–requête. La sous-requête CodedBy est utilisée pour définir les instances du 'Gene'. Cette fois, le service de données du GenBank est impliqué parce que la propriété d'objet 'CodedBy' est mappée avec le schéma XML de Genbank. La sous-requête 'ParticipateIn' est utilisée pour chercher les instances de 'Pathway'. Dans ce cas le service de données de KEGG est exécuté parce que la propriété d'objet 'ParticipateIn' est mappé avec le schéma XML de KEGG. Aussi le service de données de KEGG est impliqué en exécutant la sous-requête ForOrganism(P,O) parce que la propriété d'objet correspondante est mappée avec le schéma XML de KEGG. L'exécution du service de données de BRENDA se fait par l'utilisation de deux arguments (le numéro de classification d'enzyme et le nom de l'organisme. Pour cela, les sous-requêtes 'IsEnzyme' et 'ForOrganism' sont utilisées à la fois pour obtenir des instances de 'Enzyme'.

A chaque exécution, les services de données interrogent les sources de données, extraient les données souhaitées et retournent des documents XML. Ces résultats sont des instances des schémas XML des sources sous-jacentes. Le composant Evaluateur/intégrateur reçoit ses instances des schémas XML et, en se basant sur les règles

de correspondances entre les éléments des schémas de sources et l'ontologie de domaine enregistrés au niveau du SD-Core, les transforme en des instances de notre ontologie de domaine exprimées en RDF. Ces instances ne sont pas connectées entre elles parce qu'elles sont produites de services de données différents. Afin de les associer, l'Evaluateur/Intégrateur établie des relations entre les services de données (définis au niveau du plan de requête) et les propriétés d'objets définies au niveau de l'ontologie de domaine. Finalement, ces instances inter-reliées sont filtrées par le composant Evaluateur/Intégrateur pour éliminer toutes les informations inutiles. Le dernier résultat obtenu est une instance de l'ontologie de ***PseudmonasDW*** contenant toutes les données extraites des sources de données intégrées (Figure 31). Cette instance finale est automatiquement transformée en un document XML par l'usage de quelques librairies java (exemple Jena et Java DOM). L'étape de stockage a été réalisée automatiquement via eXist-db, où nous avons chargé tous les documents XML obtenus dans un entrepôt de données XML natif pour être interrogés via une interface utilisateur. Cette étape de stockage de données sera bien détaillée dans la section 3 du chapitre 4.


**Figure 31. Une partie de l'instance RDF de l'ontologie de domaine obtenue comme résultat final de l'étape ETL au sein de système PseudomonasDW.**

Pour résumer, nous pouvons dire que la première étape du processus ETL (Extraction) a été réalisée en utilisant les services de données pour extraire les données souhaitées à partir des sources originaux. L'étape de transformation a été partagée entre les services de données et le médiateur SB-KOM. Les services de données s'occupent par la transformation de données en format XML et le médiateur SB-KOM transforme les instances des schémas de sources en des instances exprimées en RDF afin de les intégrer dans une seule instance de l'ontologie de domaine en éliminant les redondances. La dernière étape du processus (Loading) a été réalisée par l'utilisation de eXist qui nous a permis de stocker automatiquement les données dans un entrepôt de données XML natif (Marrakchi, et al., 2010). La Figure 32 illustre toutes les étapes du processus d'ETL au sein de ***PseudmonasDW***.

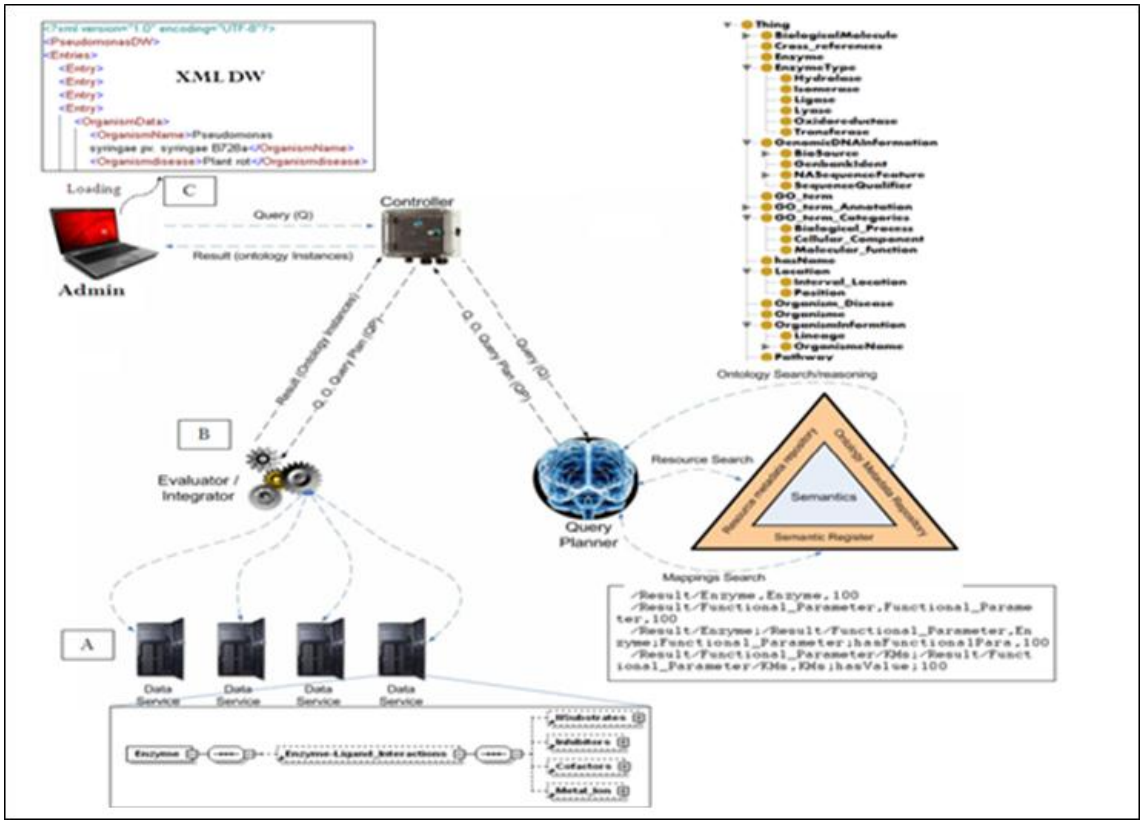


**Figure 32. Représentation schématique du processus ETL: (A) représente l'étape d'éxtraction de données, (B) représente l'étape de transformation de données et (C) représente l'étape de chargement de données au sei de PseudmonasDW.**

Les bases de données UniProt et GenBank créent des listes de diffusion. Ces listes sont destinées à la distribution des messages qui annoncent les mises à jour effectuées au niveau de ces deux bases de données. L'abonnement à ces listes nous a permis de recevoir les dernières modifications et de garder une trace des mises à jour des entrées individuelles.

Les sources de données PRODORIC, BRENDA et KEGG sont périodiquement mis à jour et fournissent des archives complètes qui contiennent uniquement les entrées actualisées. Ces archives nous ont permis de spécifier quelles entrées intégrées dans ***PseudomonasDW*** ont été mis à jour. Lorsque le système est informé par les entrées modifiées, la mise à jour des données est pratiquement intégrée à l'aide du SB-KOM.

Nous avons développé un module Java qui génère des requêtes conjonctives et les envoie au système SB-KOM pour performer les processus d'extraction et de transformation. SB-KOM fait appel aux services de Web que nous avons développé pour extraire uniquement les données modifiées à partir des entrées originales. Par la suite, il est possible de lancer automatiquement le processus d'intégration pour mettre à jour l'entrepôt de données en remplaçant seulement les données obsolètes par elles actualisées.

# 5 DISCUSSION ET CONCLUSION

L'approche entrepôt de données est née dans l'entreprise, dans les secteurs concurrentiels du commerce et du marketing. L'intérêt de l'utilisation d'une telle approche en bioinformatique s'est vite fait sentir. En effet, les atouts liés au stockage local de données et donc à l'optimisation de requête sont très adaptés aux larges volumes de données qui caractérisent les données biologiques.

Cependant, mettre en œuvre une approche entrepôt de données pour gérer et analyser des données biologiques est une tâche complexe. La nature des données que l'on doit intégrer est très différente de celle des données d'entreprise. Les données ne sont plus quantitatives mais souvent qualitatives, elles sont très nombreuses et diverses, elles sont pour la plupart réparties sur le Web, dans des sources indépendantes et très dynamiques, caractérisées par une grande hétérogénéité syntaxique et sémantique.

De ce fait, les étapes de construction de l'entrepôt n'en deviennent que plus complexes, incluant la modélisation des données biologiques ainsi que la mise en œuvre de processus d'intégration gérant la forte hétérogénéité.

La contrepartie de tous ces efforts, c'est la bonne qualité de données ensuite fournie par l'entrepôt, elle est bien souvent à l'origine de la motivation de la construction d'un tel environnement.

La quantité des données issues de l'étude biotechnologique de l'espèce de *Pseudomonas* requérant un accès à une grande diversité de données réparties dans de multiples sources de données. Nous avons donc nous-mêmes opté pour le développement d'un entrepôt de données et ainsi proposé des solutions à une intégration systématique et réconciliée de données hétérogènes.

***PseudmonasDW*** est un entrepôt de données semi-structuré pour stocker, gérer, et intégrer les informations biologiques collectées de sources de données via le Web. ***PseudmonasDW*** se focalise sur l'intégration de données de *pseudomonas sp*.

Pour la conception du système ***PseudmonasDW***, nous avons utilisé le processus d'intégration qualifié d'ascendant (ou bottom-up) où nous sommes partis du besoin de représenter au sein d'un même schéma les données souhaitées, pour ensuite choisir les sources de données ainsi que le processus d'intégration appropriés.

Ainsi, pour l'intégration de données, nous avons combiné les deux approches matérialisé et virtuelle pour exploiter leurs avantages dans un nouveau environnement hybride. Nous avons utilisé les services de données et le système médiateur SB-KOM pour extraire et intègre les données collectées à partie des sources de données. Les adaptateurs forment une partie importante dans les services de données qui fournissent des moyens pour interroger et corréler les différents types d'informations intégrés. Les services de données initialisent le processus d'ETL, dont les adaptateurs sont considérés comme une interface qui reçue des requêtes XQuery, interroge les sources de données, extraite les données souhaités et les transforme en un modèle commun utilisé par le SB-KOM.

Les différents composants du médiateur (contrôleur, planificateur de requête et l'évaluateur/intégrateur) se chargent par l'étape de transformation de données. Nous nous somme focalisés sur le développement des schémas XML pour les sources intégrés qui offrent une idée générale sur l'organisation de données au sein de sources originales. De cette manière nous avons pu développer par le biais de règles de correspondance (mappings) une intégration systématique et réconciliée des données au sein du schéma intégrateur. Comme un schéma global de l'entrepôt nous avons utilisé une ontologie de domaine qui offre une représentation formelle au monde real par la définition des concepts et des relations entre eux. Le résultat obtenu du médiateur SB-KOM est une instance de l'ontologie. L'utilisation de l'ontologie et des instances permet l'inclusion de raisonnement aux différents niveaux. Les différentes instances retournées par le SB-KOM sont chargées dans ***PseudmonasDW*** après une translation automatique en XML par le biais de quelques bibliothèques du Java. L'utilisation d'un système médiateur pour une intégration sémantique de données dans un entrepôt de données nous a permis d'exploiter leurs avantages dans une nouvelle approche. D'une part, les données sont physiquement stockées dans l'entrepôt pour être prêtes à une interrogation directe et rapide. Et d'autre part, l'intégration et la mise à jour des données sont virtuellement achevées en utilisant le médiateur.

Les différents systèmes d'intégrations développées en bioinformatique ainsi que leurs caractéristiques ont été présentés tout au long du chapitre 2. Notre approche se distingue des autres sur différents points.

Si aujourd'hui, l'environnement de ***PseudmonasDW*** permet un accès unifié à une diversité de données, l'ajout de nouvelles sources couvrant d'autre domaine de connaissance est envisageable et permettrait d'interpréter au mieux les données biologique et métabolique de *Pseudomonas sp*. Notamment, il pourrait être intéressant d'intégrer des données de puces à ADN ou encore des données d'annotation biomédicale provenant de GO.

Il faut souligner que, les entrepôts *GenMapper* ou *GeWare* sont particulièrement adaptés à l'ajout de nouvelles sources de données par l'utilisation d'un modèle générique appelé GAM. Ce dernier modélise les sources de données plutôt que leur contenu. Dans ***PseudmonasDW***, l'ajout de source supplémentaire implique une modification du schéma global. Cependant, cette modification de schéma consiste plus en une extension de schéma afin d'y ajouter de nouvelles classes permettant de décrire le domaine d'intérêt, qu'en une modification profonde du schéma.

Dans l'entrepôt *GEDAW*, la conservation de trace de données, provenant des sources intégrées, n'est pas pris en considération. Dans ce sens, la non volatilité des données caractérisant l'approche entrepôt de données n'est pas respectée. Dans notre cas, la méthode *getDataProvenance()* de services de données joue un rôle très important dans la non volatilité des données et la conservation de leur traçabilité.

Dans le cas de *BioWarehouse*, le système est linux-dépendant et exige une installation. Cela rendre l'utilisation de *BioWarehouse* une tâche laborieuse pour les biologistes qui ne maîtrisent pas l'outil informatique et particulièrement la plateforme Linux. Dans ***PseudomonasDW***, le système est plate-indépendant et n'exige aucune installation local, dont il est disponible pour l'utilisateur via une interface Web (voire chapitre 4).

Avec ***PseudmonasDW***, nous aimerions fournir aux biologistes un outil accessible pour élucider les processus cellulaire d'intérêt en utilisant une stratégie de système intégré.

# CHAPITRE 4

# PseudomonasDW et PDWiki Une plateforme biologique pour les Pseudomonas Sp.

# Chapitre 4

# PseudomonasDW et PDWiki Une plateforme biologique pour les Pseudomonas Sp.

**Sommaire**



## 1 INTRODUCTION

Les Pseudomonas forment un large groupe colonisant le sol, les plantes et l'eau. Ces bactéries Gram négatives, non sporulantes, sont aérobies obligatoires, à l'exception de certaines pouvant utiliser le NO3 comme accepteur d'électrons. Leur mobilité est assurée par plusieurs flagelles polaires, et elles ont un métabolisme mésophile et chimioorganothorphe, la plupart étant saprophytes (Emmanuel, et al., 2000). Leur facilité de culture in vitro et la disponibilité d'un nombre croissant de séquences du génome de Pseudomonas ont fait du genre Pseudomonas un foyer idéal pour la recherche scientifique.

Plusieurs bases de données de haute qualité existent déjà pour la recherche de données de séquence et des annotations pour les *Pseudomonas*, y compris le système "Integrated Microbial Genomes"[80] (IMG) (Markowitz, et al.), la ressource "JCVI Comprehensive Microbial Resource"[81] (CMR) (Peterson, et al., 2001), "xBASE"[82], "National Center for Biotechnology Information" (NCBI), "Microbial Genomes"[83] (Peterson, et al., 2001) et "Microbes Online"[84] (Glasner, et al., 2008). Bien que ces bases de données ont le but de faciliter la recherche et la comparaison des annotations génomiques sur la gamme complète des procaryotes, mais aucune met l'accent sur une curation interne pour les Pseudomonas (Winsor, et al., 2009). Autres bases de données telles que "Enteropathogen Resource Integration Center"[85] (McLeod, et al., 2006) et le site "Pseudomonas syringae Genome Resources"[86] se focalisent sur la maintenance d'une grande qualité de curation pour un groupe taxonomique spécifique tout en mettant l'accent sur le suivi des changements des annotations et de permettre leur comparaison entre les espèces et les souches de leurs groupes respectifs (Winsor, et al., 2009). D'autre part, "Pseudomonas Genome Database"[87] (Winsor, et al., 2009) est une des bases de données fameuses qui s'intéressent à l'annotation des génomes des Pseudomonas. Cette base de données se focalise sur l'annotation du génome de Pseudomonas aeruginosa PAO1 et fournit des informations pertinentes pour la recherche génomique de cette espèce mais manque de données reliées à la protéine et aux autres concepts biologiques comme les voies métaboliques et les réactions enzymatiques. Pour les autres souches de Pseudomonas, la base de données "Pseudomonas Genome Database" offre un ensemble de données qu'on peut le considérer pauvre par rapport aux données relatives au Pseudomonas aeruginosa PAO1.

Dans ce chapitre nous présentons le produit de l'approche hybride décrit dans le chapitre précédent : ***PseudomonasDW*** un entrepôt de données semi-structuré qui regroupe des données génomiques, protéiques, enzymatiques et métaboliques de l'espèce de Pseudomonas. ***PseudomonasDW*** incorpore 33 bases de données natives chacune pour une espèce ou une souche de Pseudomonas sp. Dans ce chapitre nous détaillons la phase de l'implémentation de ces bases de données en décrivant leur contenu, la manière de les accéder et de naviguer. ***PseudomonasDW*** est prolongé par un wiki biologique spécifique aux espèces de Pseudomonas nommé ***PDWiki*** qui donne à l'utilisateur de ***PseudomonasDW***, l'occasion d'ajouter et d'éditer des informations supplémentaires concernant les espèces de Pseudomonas.

[80] http://img.jgi.doe.gov
[81] http://cmr.jcvi.org/tigr-scripts/CMR/CmrHomePage.cgi
[82] http://www.xbase.ac.uk/
[83] http://www.ncbi.nlm.nih.gov/genomes/MICROBES/microbial_taxtree.html
[84] http://www.microbesonline.org/
[85] http://patricbrc.vbi.vt.edu/portal/portal/patric/IncumbentBRCs?page=eric
[86] http://www.pseudomonas-syringae.org/
[87] http://www.pseudomonas.com/

# 2 MODÉLISATION DE PSEUDOMONASDW

Il est bien connu qu'avant d'entreprendre la réalisation informatique d'un "problème", il est nécessaire de réfléchir aux tenants et aboutissants du système à réaliser : il s'agit de passer du monde réel, complexe et confus, au monde informatique où les structures et les propriétés des objets doivent être identifiées. Cette tâche classique est également essentielle dans la modélisation d'une base de données. Cette phase de modélisation nécessite de nombreux choix qui auront des répercussions importantes dans la suite.

La modélisation se réalise en trois étapes principales qui correspondent à trois niveaux d'abstraction différents:

- **Modèle conceptuel** : représente le contenu de la base en termes conceptuels, indépendamment de toute considération informatique.
- **Modèle logique** : résulte de la traduction du schéma conceptuel en un schéma propre à un type de base de données.
- **Modèle physique** : est utilisé pour décrire les méthodes d'organisation et d'accès aux données de la base.

La modélisation conceptuelle est une étape fondamentale de la conception des systèmes informatiques. Elle a pour objectif une prise en compte plus adéquate des besoins des applications dans leur environnement d'utilisation. La modélisation conceptuelle consiste à représenter de manière abstraite, c'est-à-dire en termes de concepts familiers aux domaines d'application et indépendamment des technologies d'implémentation, certains aspects des systèmes physiques ou humains et de leur environnement.

Toute la modélisation conceptuelle de l'entrepôt ***PseudomonasDW*** a été effectuée grâce aux différents diagrammes proposés par la méthodologie UML[88] (Unified Modelling Language voir Annexe 1). Nous avons choisi le langage UML pour ses caractéristiques et son dynamisme permettant une modélisation aisée des problèmes, entre autres, biologiques et bioinformatiques. Nous n'avons pas la prétention de présenter ci-dessous un tutorial sur l'UML. Seulement, nous nous mettrons d'accord sur les acquis fondamentaux fournis par ce langage pour la conception de ***PseudomonasDW***.

## 2.1 Diagrammes des cas d'utilisation du système PseudomonasDW

Le digramme des cas d'utilisation représente l'ensemble des cas d'utilisation de ***PseudomonasDW*** (Un cas d'utilisation est une unité cohérente représentant une

[88] Vous pourriez vous référer à [http://www.uml.org/] pour une étude de ce langage.

fonctionnalité visible de l'extérieur.), les acteurs en jeu (Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système.) et les relations entre ces différents cas. Il capture le comportement du système tel qu'un utilisateur extérieur le voit.

Notre système présent pour l'instant trois acteurs (**Table 4**) que sont l'administrateur (ou le bioinformaticien), l'entrepôt de données **PseudomonasDW** et l'utilisateur (ou le biologiste).

**Table4 : La liste des acteurs.**

| Acteur | Cas d'utilisation |
|---|---|
| Utilisateur | Un interlocuteur interconnecté avec le système via internet |
| PseudomonasDW | Le système avec lequel l'utilisateur se connecte via une interface web |
| Administrateur | Le superviseur du système |

**L'utilisateur** peut interroger l'entrepôt de données en envoyant des mots clés via l'interface Web ; comme il peut analyser les données en utilisant les fonctionnalités fournies par le système. Les principales opérations de l'utilisateur sont définies comme suit :

- L'utilisateur demande une connexion au système ***PseudomonasDW*** en introduisant son URL.
- L'utilisateur interroge le système ***PseudomonasDW*** en introduisant des mots clés via son interface web.
- L'utilisateur analyse les données fournies par ***PseudomonasDW*** en utilisant les différentes fonctionnalités du système.

### *a) Liste des cas d'utilisation de l'utilisateur* (Table5)

**Table5 : les cas d'utilisation de l'utilisateur.**

| Cas d'utilisation |
|---|
| Etablissement d'une connexion avec le système. |
| Interrogation du système. |
| Analyse de données. |

b) ***Le diagramme de cas d'utilisation de l'utilisateur*** (Figure 33)

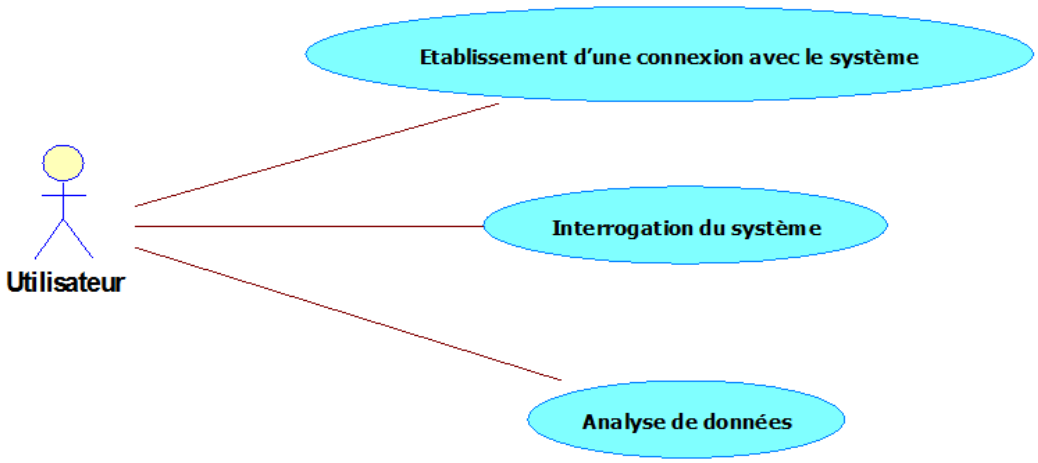


**Figure 33. Le diagramme de cas d'utilisation de l'utilisateur**

**PseudomonasDW** offre une interface web entre l'utilisateur et l'ensemble de données stockées au niveau de l'entrepôt de données. Les principales opérations du ***PseudomonasDW*** sont comme suit :

- Translation de la requête par l'utilisation des mots clés introduits par l'utilisateur pour la constitution d'une requête convenable au schéma du système.
- Construction du résultat
- Translation du résultat en un format lisible par l'utilisateur.

***a) Liste des cas d'utilisation de PseudomonasDW*** (Table6)

**Table 6 : les cas d'utilisation de PseudomonasDW.**

| Cas d'utilisation |
|---|
| Translation de la requête. |
| Construction du résultat |
| Translation du résultat |

### b) *Le diagramme de cas d'utilisation de PseudomonasDW* (Figure 34)

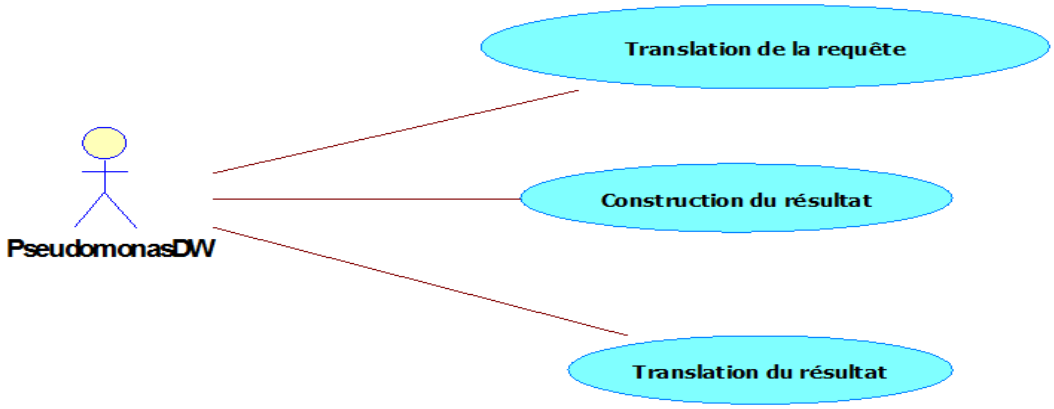


**Figure 34. Le diagramme de cas d'utilisation de PseudomonasDW**

**L'administrateur** est le superviseur du système. Il interagit avec l'entrepôt pour intégrer, nettoyer et rafraîchir (mettre à jour) les données. Il intervient également pour réaliser l'interface de l'entrepôt et y rajouter des fonctionnalités lorsque les biologistes en émettent le souhait. Les principales opérations de l'utilisateur sont comme suit :

- Intégration de données au sein de ***PseudomonasDW***
- Nettoyage de données en éliminant les redondances
- Mise à jour de données par l'ajout, la suppression et la modification de données en fonction des sources originales
- Maintenance de l'entrepôt de données
- Maintenance de l'interface Web
- Ajout des fonctionnalités en cas de besoin

### a) *Liste des cas d'utilisation de l'administrateur* (Table7)

**Table 7 : les cas d'utilisation de l'administrateur.**

| Cas d'utilisation |
|---|
| Intégration de données |
| Nettoyage de données |
| Mise à jour de données |
| Maintenance de *PseudomonasDW* |
| Maintenance de l'interface Web |
| Ajout de fonctionnalités |

***b) Le diagramme de cas d'utilisation de l'administrateur*** (Figure 35)

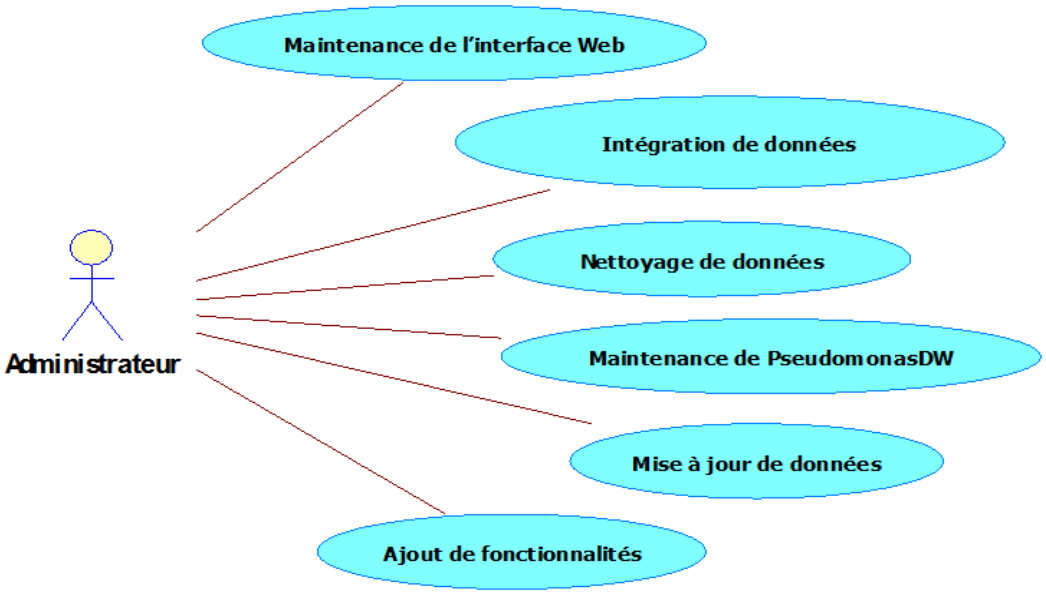


**Figure 35. Le diagramme de cas d'utilisation de l'administrateur**

## 2.2 Diagrammes de séquence du système PseudomonasDW

Les diagrammes de séquences permettent de représenter des collaborations entre les objets selon un point de vue temporel. Ils sont, en général, utilisés pour modéliser les aspects dynamiques des systèmes en temps réel. Les diagrammes de séquences ont été désignés sous plusieurs noms, dont diagrammes d'interactions, tracé de messages, ou tracé d'événements. Leur notation est dérivée principalement du '*Object Message Sequence Chart*' du Siemens Pattern Group (Buschmann, et al., 1996).

Le diagramme de séquence ci-dessous (Figure 36) représente des événements et des messages envoyés lors de l'interrogation des bases de données de ***PseudomonasDW*** (PDW D.B) par un utilisateur via l'interface Web (Web app). **La Table 8** résume les différents messages envoyés en indiquant pour chaque message son émetteur et son récepteur.

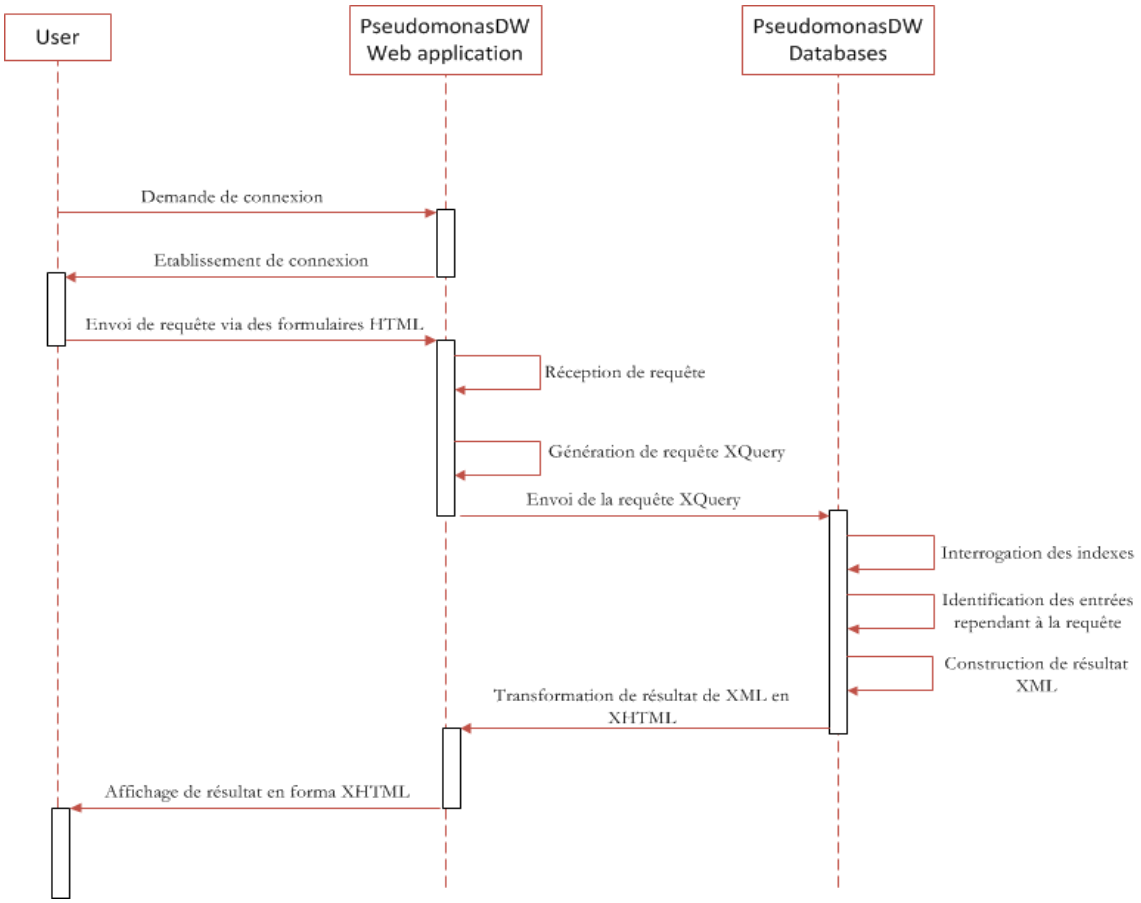


**Figure 36. Le diagramme de séquence: interrogation de PseudomonasDW par l'uilisateur**

**Table8 : La liste des mesages envoyés entre l'utilisateur, l'interface Web et les bases de données de PseudomonsDW.**

| message | | émetteur | récepteur |
|---|---|---|---|
| 1. | Demande de connexion | Utilisateur | Web app |
| 2. | Etablissement de connexion | Web app | Utilisateur |
| 3. | Envoi de requête via des formulaires HTML | Utilisateur | Web app |
| 4. | Réception de requête | Web app | Web app |
| 5. | Génération de requête XQuery | Web app | Web app |
| 6. | Envoi de la requête XQuery | Web app | PDW D.B |
| 7. | Interrogation des indexes | PDW D.B | PDW D.B |
| 8. | Identification des entrées rependant à la requête | PDW D.B | PDW D.B |
| 9. | Construction de résultat XML | PDW D.B | PDW D.B |
| 10. | Transformation de résultat de XML en XHTML | PDW D.B | Web app |
| 11. | Affichage de résultat en forma XHTML | Web app | Utilisateur |

### 2.3 Diagramme de classes du système PseudomonasDW

Le diagramme de classes (Figure 37) constitue un élément très important de la modélisation de ***PseudomonasDW***, il nous a permis de définir quelles seront les composantes du système final : il est considéré comme une représentation statique des éléments qui composent les bases de données de ***PseudomonasDW*** et de leurs relations. Nous nous sommes basés sur les données proposées par les sources intégrées et les différents concepts de l'ontologie de ***PseudomonasDW***, préalablement développé lors de la phase d'intégration de données (voir la section 3.3 du chapitre précédent), pour définir les différentes classes et relations composant notre diagramme de classe.

Le diagramme de classe de ***PseudomonasDW*** est constitué de six classes principales (classe '*Genome*', classe '*Gene*', classe '*Protein*', classe '*Enzyme*' et la classe '*Pathway*') auxquelles ont été ajoutées d'autres classes qui donnent plus de spécialisation et de raffinement au modèle conceptuel du système. Par conséquent, le modèle conceptuel nous a permis de mieux comprendre la structure de ***PseudomonasDW***, ainsi que de décrire ses différents concepts et les relations qui les lient. Les classes représentent les modules des bases de données de ***PseudomonasDW***, elles sont représentées par des rectangles divisés en trois sections : la section supérieure contient le nom de la classe, la section centrale définit les propriétés de la classe et la section du bas énumère les méthodes de la classe. Les différentes classes du notre modèle conceptuel sont reliées par des relations d'association qui sont modélisées par des lignes reliant deux classes, des relations de spécialisation qui sont représentées par des flèches allant de la sous classe à la super classe et des relations de composition qui sont représentées par des lignes avec un losange à la base.

# 3 IMPLEMENTATION DE PSEUDOMONASDW

Comme nous avons déjà mentionnés tout au long de ce manuscrite, l'objectif de cette thèse est la mise en place d'un entrepôt de données XML spécifique aux espèces de Pseudomonas. Les entrepôts de données XML forment une base intéressante pour les applications décisionnelles qui exploitent des données hétérogènes et provenant de sources multiples.

Les travaux menés dans le contexte de l'entreposage de données XML peuvent être divisés en deux familles (Mahboubi, et al., 2009).

- La première famille propose une modélisation multidimensionnelle pour les entrepôts de données XML. Elle se base sur les modèles classiques (schémas en étoile et dérivés). Ces travaux permettent ainsi une utilisation dynamique des dimensions et offrent un support pour des outils d'analyse.

- Les approches de la seconde famille abordent la problématique de l'entreposage de documents XML. Elles perçoivent un entrepôt XML comme une collection de documents XML.

Pour le développement des bases de données de ***PseudomonasDW***, nous nous sommes basés sur les approches de la deuxième famille, où nous avons incorporés les données, extraites à partir des sources de données intégrées, dans des documents XML. Chacun d'eux étant stocké dans une collection de documents XML.

Nous nous sommes arrêtés dans la section 4 du chapitre 3 au point du stockage des documents XML, obtenus de la transformation des instances RDF, au niveau de notre entrepôt de données ***PseudomonasDW***. Dans les sous-sections suivantes nous comptons donner une vue générale sur le processus de stockages des documents XML dans les bases de données et la manière de leur implémentation. Nous avons utilisé les bases de données XML natives (voir Annexe 2) et principalement le logiciel libre eXist (voir Annexe 3).

## 3.1 Organisation des bases de données de PseudomonasDW

Actuellement ***PseudomonasDW*** contient des informations concernant 33 espèces du genre Pseudomonas (**Table 9**) stockées dans 33 bases de données XML natives (une base de données pour chaque espèce). Une base de données est représentée par une collection des documents XML où nous avons déjà stockés les données. Les données sont structurées selon un schéma XML (modèle logique de données) obtenue par la réconciliation des schémas XML des sonurces de données définies dans la section 3.1 du chapitre 3. Ce modèle de données définie l'organisation et la restriction de données dans chaque entrée de l'entrepôt. Nous avons considéré que chaque document XML est une entrée de ***PseudomonasDW*** identifiée par un numéro d'accession unique. Pour cela, nous avons nommées l'élément racie, du modèle de données, « *Entry* ».

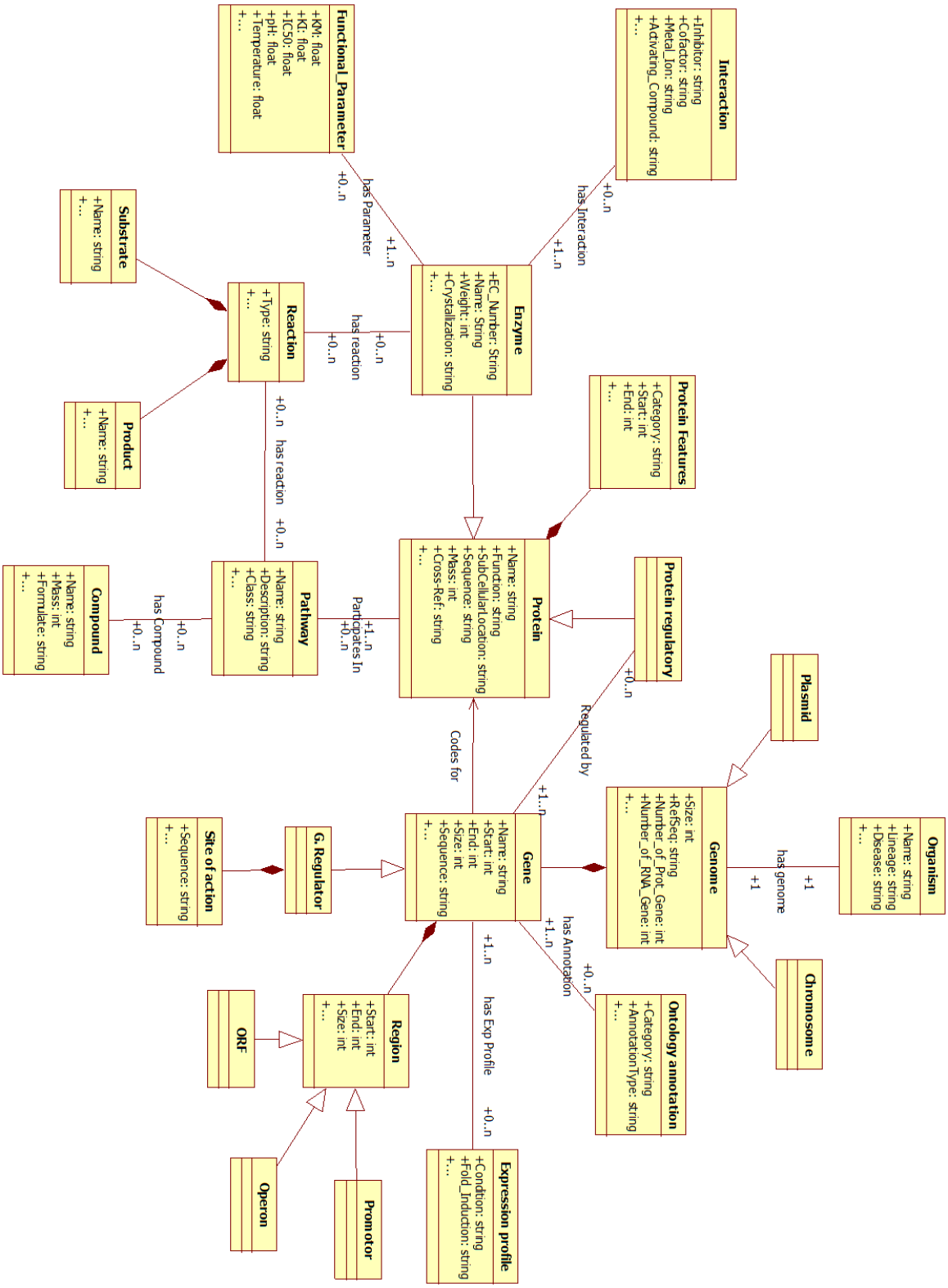


**Figure 37. Le diagramme conceptuel de PseudomonasDW**

**Table9 : Quelques statistiques concernant les espèces de Pseudomonas intégrées dans PseudomonasDW.**

| Pseudomonas Sp | Taille de genome (bp) | Nombre des gènes | Nombres des entrées |
|---|---|---|---|
| **Genomes complets** | | | |
| Pseudomonas aeruginosa PAO1 | 6,264.404 | 5,682 | 5,556 |
| Pseudomonas aeruginosa M18 | 6,327.754 | 5,764 | 5,684 |
| Pseudomonas aeruginosa NCGM2.S1 | 6,764.661 | 6,538 | 6,269 |
| Pseudomonas aeruginosa LESB58 | 6,601.757 | 6,061 | 5,908 |
| Pseudomonas aeruginosa PA7 | 6,588.339 | 6,369 | 6,246 |
| Pseudomonas aeruginosa UCBPP-PA14 | 6,537.648 | 5,977 | 5,886 |
| Pseudomonas fluorescens PfO-1 | 6,438.405 | 5,829 | 5,714 |
| Pseudomonas fluorescens Pf-5 | 7,074.893 | 6,233 | 6,137 |
| Pseudomonas fluorescens SBW25 | 6,722.539 | 6,106 | 5,921 |
| Pseudomonas fluorescens F113 | 6,845.832 | 5,953 | 5,862 |
| Pseudomonas putida F1 | 5,959.964 | 5,403 | 5,245 |
| Pseudomonas putida GB-1 | 6,078.430 | 5,529 | 5,408 |
| Pseudomonas putida KT2440 | 6,181.863 | 5,516 | 5,350 |
| Pseudomonas putida W619 | 5,774.330 | 5,309 | 5,182 |
| Pseudomonas putida BIRD-1 | 5,731.541 | 5,046 | 4,960 |
| Pseudomonas putida S16 | 5,984.790 | 5,307 | 5,171 |
| Pseudomonas syringae pv.phaseolicola | 6,112.448 | 5,437 | 5,172 |
| Pseudomonas syringae pv.tomato | 6,397.126 | 5,688 | 5,481 |
| Pseudomonas syringae pv.syringae | 6,093.698 | 5,220 | 5,089 |
| Pseudomonas stutzeri A1501 | 4,567.418 | 4,210 | 4,128 |
| Pseudomonas stutzeri DSM 4166 | 4,689.946 | 4,372 | 4,301 |
| Pseudomonas stutzeri ATCC 17588 | 4,547.930 | 4,287 | 4,181 |
| Pseudomonas entomophila L48 | 5,888.780 | 5,275 | 5,134 |
| Pseudomonas mendocina ymp | 5,072.807 | 4,704 | 4,594 |
| Pseudomonas mendocina NK-01 | 5,434.353 | 5,035 | 4,954 |
| Pseudomonas brassicacearum NFM421 | 6,843.248 | 6,176 | 6,081 |
| Pseudomonas fulva 12-X | 4,920.769 | 4,540 | 4,459 |
| **Genomes incomplets** | | | |
| Pseudomonas aeruginosa C3719 | ≈ 6,146.998 | 5,626 | 5,207 |
| Pseudomonas aeruginosa 2192 | ≈ 6,826.253 | 6,243 | 5,905 |
| Pseudomonas aeruginosa 152504 | ≈ 6,813.259 | 6,499 | 6,221 |
| Pseudomonas aeruginosa 138244 | ≈ 6,357.409 | 6,230 | 6,096 |
| Pseudomonas aeruginosa 39016 | ≈ 6,866.064 | 6,468 | 6,402 |
| Pseudomonas chlororaphis | - | - | 218 |

Toutes les bases de données de ***PseudomonasDW*** sont centralisés sur cinq concepts (ou entités biologiques) (Figure 38): Organisme, Gène, Protéine, Enzyme et voie métabolique. Ces concepts sont représentés, dans le modèle de données, par cinq éléments figurés directement après l'élément racine.

- L'élément «*OrganismData*» et ses descendants décrivent les données et leur organisation reliées à l'espèce de Pseudomonas de la base de données correspondante.
- L'élément «*GeneData*» est créé pour encapsuler et modéliser les données reliées au gène codant à la protéine décrite au niveau de l'entrée.
- Les données reliées directement à la protéine décrite par une entrée sont structurées sous l'élément «*ProteinData*».

- Plusieurs enzymes éventuelles peuvent être reliées à une seule protéine dans ***PseudomonasDW***. L'élément « *EnzymeData*» est un élément optionnel qui compte définir et organiser les données concernant les enzymes et leurs propriétés.
- Le dernier fils de l'élément « *Entry* » est l'élément «*PathwayData*» qui détermine les différentes voies métaboliques dans lesquelles participe la protéine définit dans l'entrée.


**Figure 38. L'organisation de données dans les bases de données de PseudomonasDW. A gauche, les cinqs éléments du niveau le plus haut du modèle de données de PseudomonasDW. A droite, un exemple d'un document XML stocké dans la base de données de Pseudomonas aeruginosa PAO1.**

## 3.2 Implémentation des bases de données de PseudomonasDW

En général, ***PseudomonasDW*** utilise les deux technologies JAVA et XML. Les données sont stockées dans des bases de données XML natives selon le modèle de données XML décrit dans la section précédente 3.2. Les bases de données natives sont gérées par la version eXist-db 1.4.0. Nous avons utilisé eXist comme étant une distribution autonome qui s'exécute à l'intérieur d'une application Web servis par un serveur préconfiguré nommé *Jetty*[89], cela nous a permis de bénéficier de toutes ses interfaces, utilisées comme des servlets, pour l'accès distant.

[89] http://jetty.codehaus.org/jetty/

La fenêtre « Client d'administration » (Figure 39) fournit par eXist, nous a permis de charger automatiquement (en utilisant les différentes options du menu) les documents XML dans 33 collections : une collection pour chaque espèce entreposé dans ***PseudomonasDW***. L'interrogation des collections a été effectuée à partir de notre application Java via l'API XML:DB[90]. Le langage de requête utilisé est le standard XQuery. Le processus de requête est extensible et dispose d'une vaste collection de module de fonctions de XQuery.

Dans le but de faciliter et d'accélérer le processus d'interrogation des bases de données de ***PseudomonasDW***, nous avons développé des indexes qui sont créés et maintenus automatiquement dans eXist. Nous avons suivis la nouvelle procédure d'indexation basée sur les noms des éléments. Cela nous a permis de retrouver facilement tous les éléments d'un certain nom quelle que soit leur imbrication.

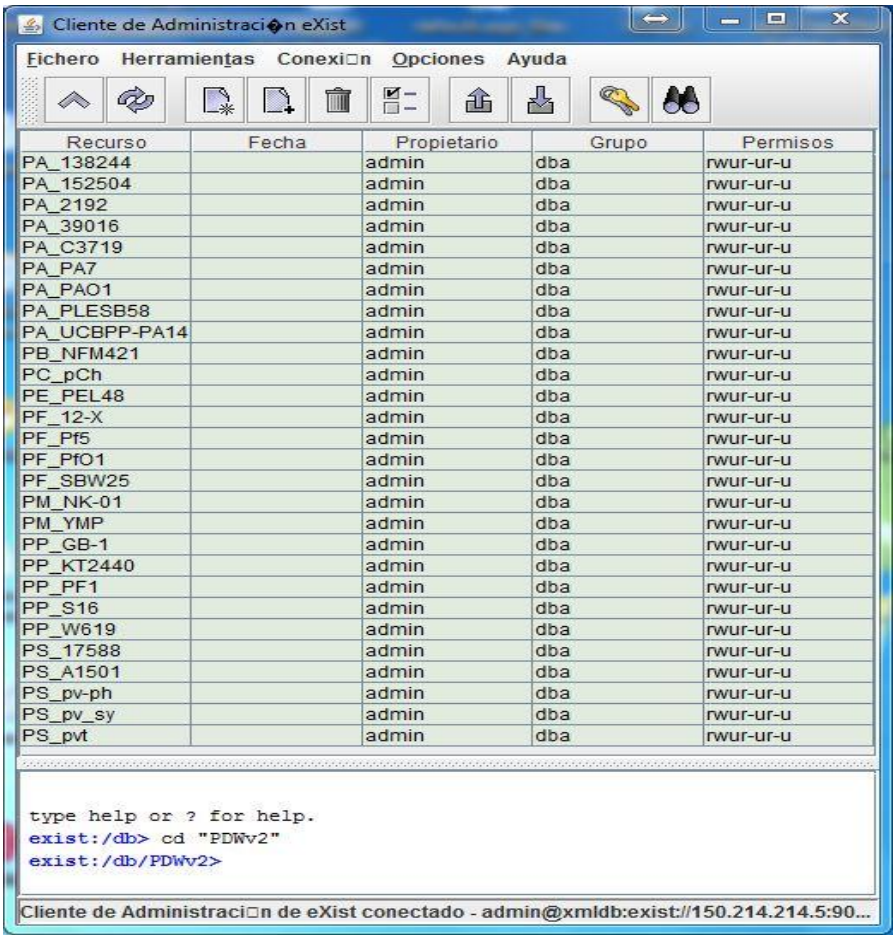


**Figure 39. La fenêtre "Client d'administration d'eXist" représentant les 33 collections stockées au niveau de PseudomonasDW**

[90] XML:DB : API qui propose une interface pour l'accès aux bases de données natives ou toute autre base de données supportant XML.

# 4 INTERFACE WEB DE PSEUDOMONASDW

Les bases de données de ***PseudomonasDW*** sont publiquement accessibles via une interface Web disponible sur le lien http://www.pseudomonasdw.khaos.uma.es . C'est une application web que nous avons développé en utilisant principalement quelques technologies du Web et de Java (JSP, Java, Servlet API, XHTML, CSS, XSLT, JavaScript, JQuery). L'application Web est implémentée sur le serveur Web *Apache 2.0*.

## 4.1 Les Moteurs de recherché dans PseudomonasDW

L'interface Web de ***PseudomonasDW*** propose deux formulaires de recherche ou des moteurs de recherche pour accéder aux données stockées au niveau des bases de données XML natives.

**Le formulaire simple ou rapide (**Figure 40**)** : il apparut en haut de toutes les pages de l'interface Web et permet d'envoyer rapidement les requêtes en se basant sur quelques mots clés (Nom du gène ou de Protéine, terme de GO ou n'importe quel mot clé qui apparut dans les champs de recherche des bases de données intégrées). Le moteur de recherche rapide offre la possibilité de restreindre la recherche en utilisant une option de recherche qui permet à l'utilisateur de sélectionner une espèce spécifique de Pseudomonas parmi l'ensemble des espèces intégrées (Figure 41). Le formulaire offre aussi un menu « *drop-down* » (Figure 42) avec lequel l'utilisateur peut limiter sa recherche dans un champ spécifique. Par exemple, l'utilisateur peut sélectionner « *Protein Names* » dans le menu « *drop-down* » pour orienter la recherche seulement dans les champs où figurent les noms de la protéine et ignorer tous les autres champs. Cette option nous a permis d'aider l'utilisateur à minimiser le temps et la complexité de la recherche.

**Le moteur de recherche avancé (**Figure 43**):** ce dernier offre à l'utilisateur la possibilité de soumettre des requêtes complexes basées sur plusieurs mots clés. Ce formulaire de recherche ou moteur de recherche propose des champs de recherche multiple où l'utilisateur peut spécifier des mots clés reliés aux différentes données de Pseudomonas stockées au niveau des bases de données ( Sub-cellular Location, Protein Existence, Operon, Gene Ontology Term, EC Number, Pathway Name, etc). Nous avons aussi équipé ce formulaire de recherche avec une option pour choisir une ou plusieurs espèces pour la reconstitution de la requête. De cette manière, les utilisateurs ont la possibilité de soumettre des requêtes en même temps à plusieurs bases de données. Autrement dit, les utilisateurs peuvent chercher dans un nombre de bases de données allant de 1 à 33.

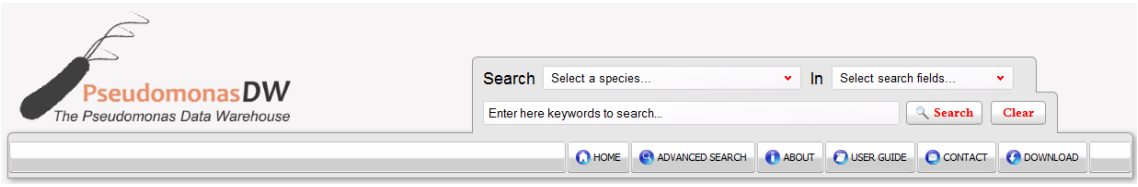


**Figure 40. Le moteur de recherche rapide ou (Simple) de l'interface Web de Pseudomonas.**

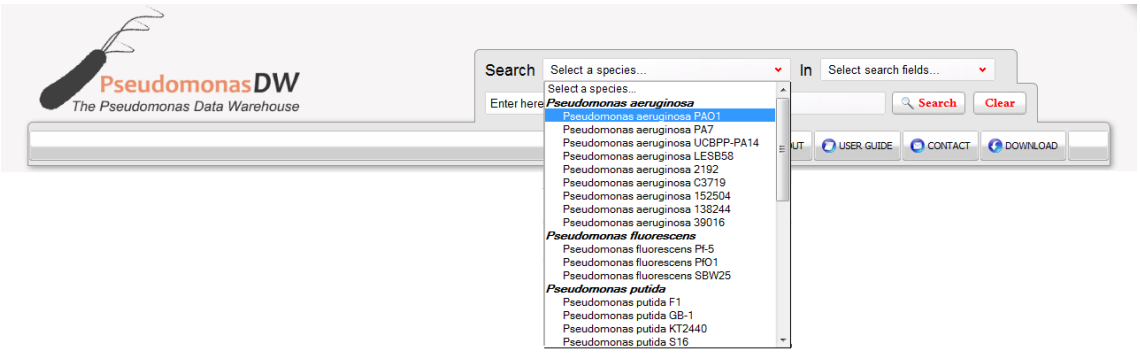


**Figure 41. Une capture d'écran de l'un des champs du moteur de recherche rapide qui donne la possibilité de sélectionner l'espèce souhaité.**

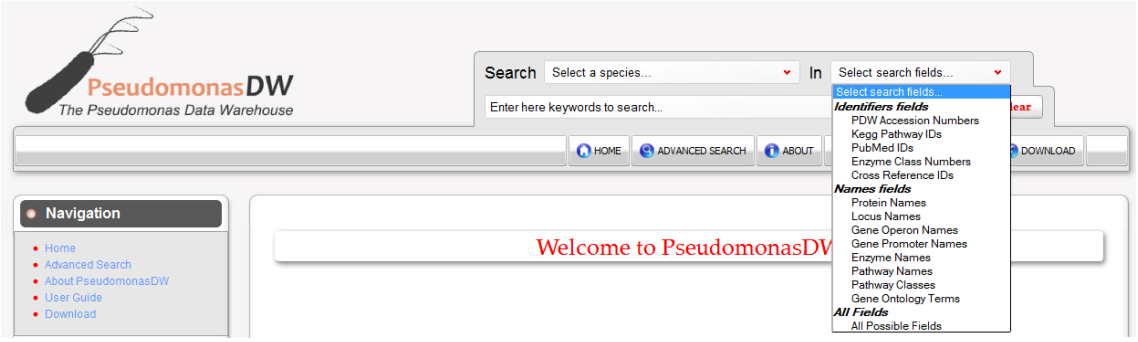


**Figure 42. Une capture d'écran du menu "drop-down" qui offre à l'utilisateur la possibilité de sélectionner un champ spécifique de recherche**

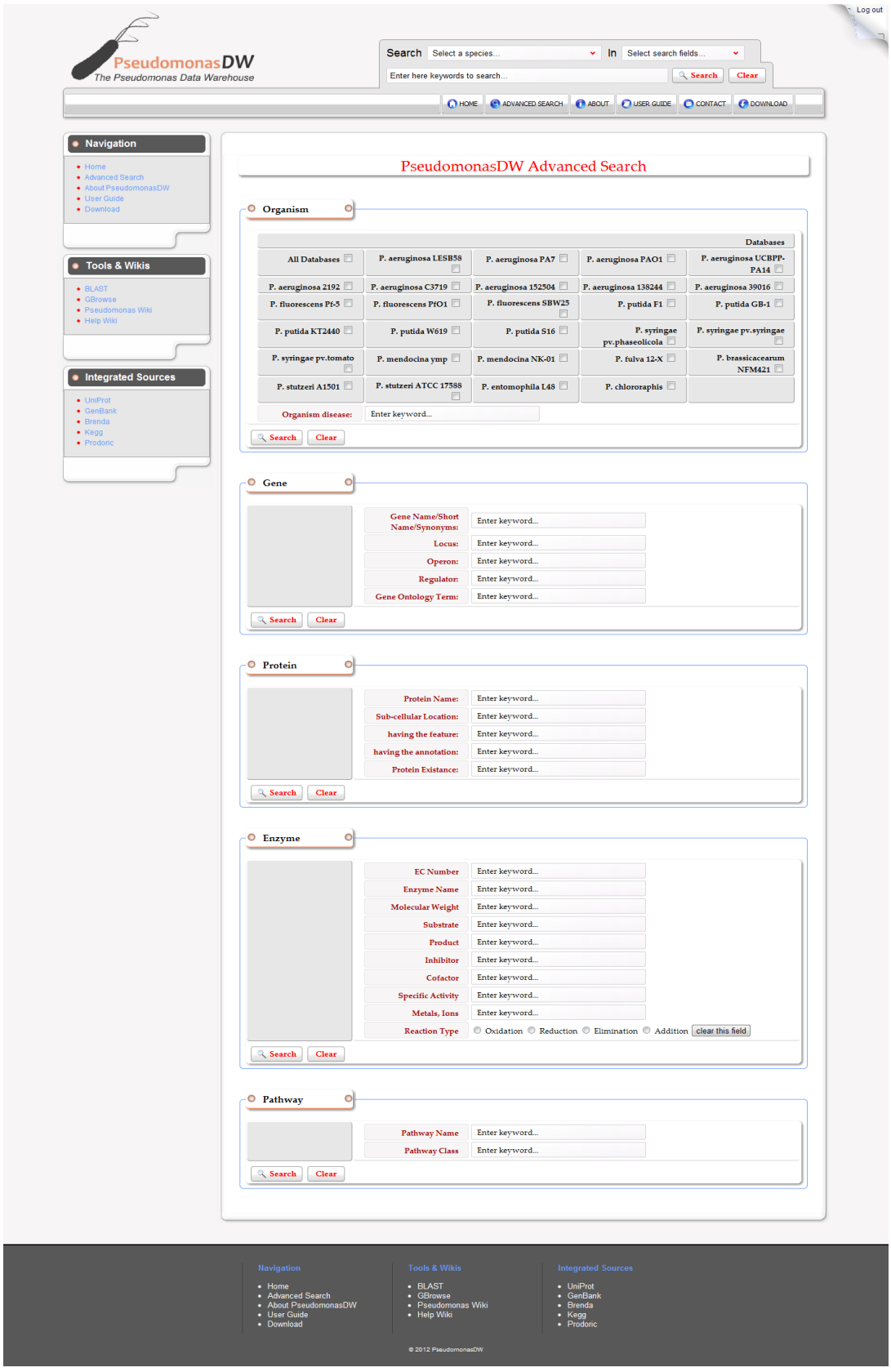


**Figure 43. Une capture d'écran de la page Web du moteur de recherche avancé**

Chaque formulaire de recherche (rapide et avancé) utilise une servlet distingue nommée «*Post method*». Ces servlets reçoivent des mots clés spécifiques et faites appel à quelques classes Java qui génèrent des requêtes XQuery pour être envoyer aux bases de données de ***PseudomonasDW***. L'application Web reçoivent des repenses de format XML et utilisent quelques feuilles de styles (XSLT et CSS) pour convertir ces repenses à des vues HTML montrant toutes les entrées correspondantes à la requête. Un effort considérable a été aussi investi pour rendre la recherche dans ***PseudomonasDW*** assez simple et convenable pour les utilisateurs qui n'ont pas une connaissance détaillée aux données de ***PseudomonasDW***. Le site Web offre aussi la possibilité de télécharger des données dans quelques formats qui dépendent à l'ensemble de données choisis :

- Un ensemble d'entrées est téléchargeable en format XML.
- Des séquences nucléiques et d'acides aminés sont téléchargeables en format Fasta.
- Quelques annotations de séquences sont téléchargeables en formats GFF3.

## 4.2 Les entrées de Pseudomonas DW

Chaque entrée de ***PseudomonasDW*** (Figure 44) décrie une protéine donnée selon cinq sections (suivant les cinq éléments principaux du modèle de données XML définit dans la section 3.1.3) : '*Organism'*, '*Gene'*, '*Protein'*, '*Enzyme'* et '*Pathways'*. Toutes ces sections sont listées dans une seule page HTML. Une barre de menu dynamique, facilite le passage d'une section à autre par un simple clic, est situé au haut de chaque page d'entrée. Les entrées de ***PseudomonasDW*** listent des informations utiles qui sont décrit d'une manière détaillée dans la page '*User guide*' qui est disponible en ligne sur le site Web. Ci-après quelques détails des cinq sections :

La section '*Organism* décrit les informations reliées à l'espèce sous-jacent à l'entrée. Ces informations concernent principalement le nom de l'organisme, sa taxonomie, le type et la langueur du chromosome, plus de quelques statistiques sur le nombre des gènes codant pour les protéines et les ARN.

La section '*Gene'* cite des informations reliées au gène codant pour la protéine en question. Les données de cette section offrent une brève description du gène, le nom scientifique, les références bibliographiques et une table de caractéristiques décrivant les différents domaines biologiques du gène. Ces derniers incluent les régions codantes de la séquence nucléotidique, les ORFs, les Operons, les Promoteurs, les facteurs de transcriptions, les sites de liaison, et les sites de mutations ou de modification. Cette section offre aussi les coordonnés chromosomiques et la séquence nucléotidique. Une image du gène générée par l'outil *GBrouse* (Donlin, 2002) est aussi représentée dans cette section. A partir de l'image de *GBrowse*, l'utilisateur peut naviguer à l'outil en cliquant sur l'image.

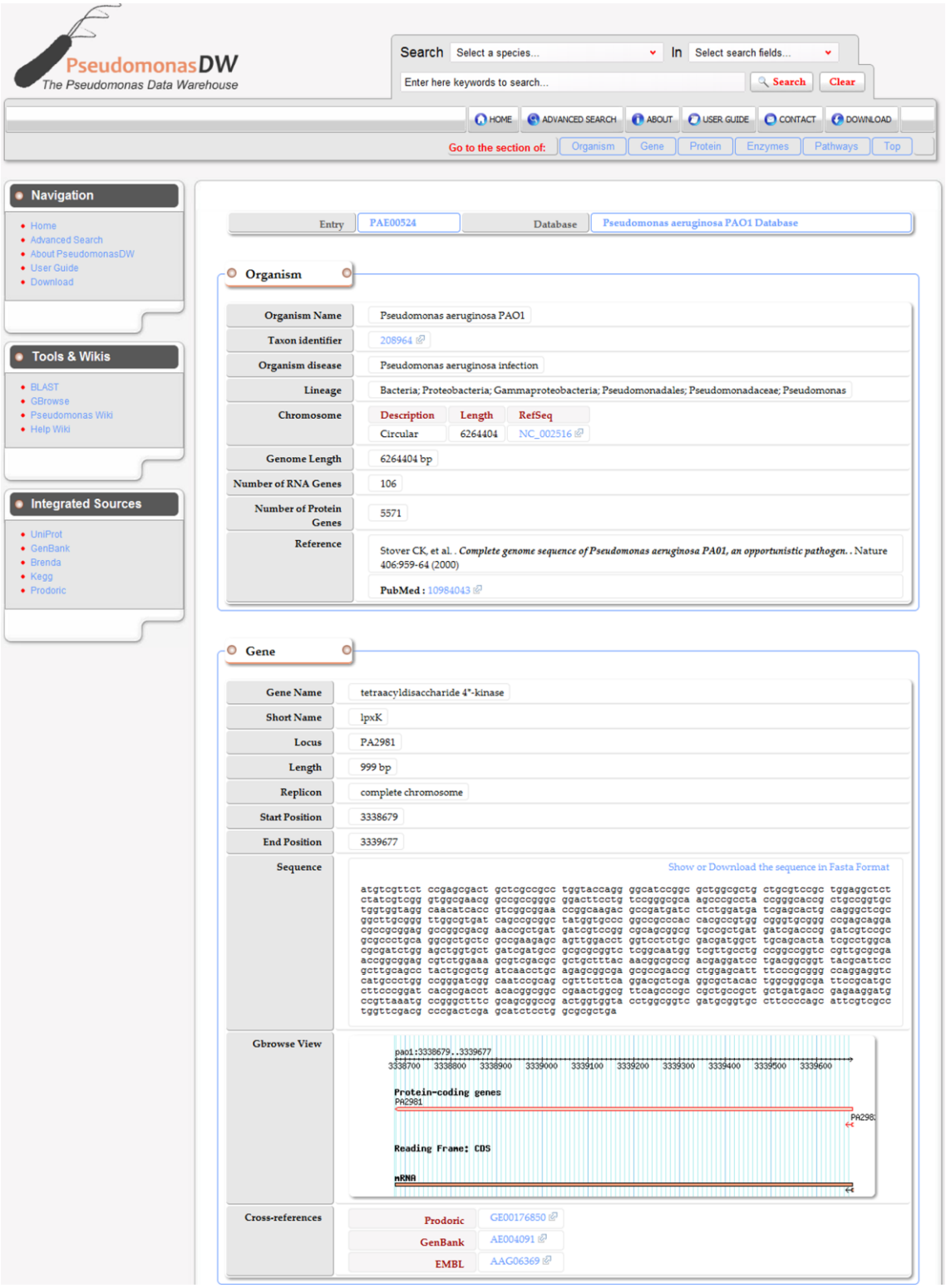


**Figure 44. Un exemple de l'entrée de PseudomonasDW, il représente les deux sections 'Organism' et 'Gene' de l'entrée PAE00524**

La section '*Protein'* présente des informations sur la protéine décrite dans l'entrée. Elle contient souvent une large quantité de données qui doit être représentée d'une manière qui permet un affichage et une lecture très simple. Les informations de cette section sont représentées dans des tableaux concernant, en plus de la nomenclature scientifiques de la protéine, la fonctionnalité de la protéine, l'activité catalytique, le mécanisme de régulation et l'annotation de 'Gene Ontology'. La section '*Protein'* liste aussi les différentes caractéristiques de la protéine (les sites de liaisons, les chaines, les hélix … etc), les références bibliographiques, des cross-références vers d'autres bases de données ainsi que la séquence peptidique de la protéine.

La section '*Enzyme'* offre des informations sur les activités enzymatiques de la protéine décrite dans l'entrée. Cette section offre les informations suivantes : 'Enzyme Commission number', ce numéro a un lien direct vers l'entrée correspondante dans la base de données enzymatique *Brenda*, la nomenclature de l'enzyme et une brève description des réactions catalytique auxquelles elle participe (le nom et le type de la réaction, les noms des substrats et des produits en plus de quelques commentaires). La section '*Enzyme'* offre aussi des informations sur les interactions enzyme_ligand impliquant l'enzyme décrite. En plus des informations sur la structure de l'enzyme, quelques propriétés moléculaires et des paramètres fonctionnels sont aussi représentés par la section '*Enzyme'*.

La section '*Pathway'* décrit les informations sur toutes les voies métaboliques dans lesquelles participe la protéine décrite dans l'entrée. Ces informations sont principalement propagées vers le nom de la voie métabolique, le numéro d'accession dans la base de données KEGG, les classes de la voie métabolique (par exemple la classe métabolisme …), l'ensemble des protéines et les composants chimiques qui participent dans la voie métabolique. La section '*Pathway'* offre une image statique pour chaque voie métabolique présenté dans l'entrée ; cette image offre une représentation graphique de tous les composants et les modules de la voie métabolique.

Les deux sections '*Organism'* et '*Protein'* sont des sections permanentes dans toutes les entrées de ***PseudomonasDW***. Les autres sections sont optionnelles selon la présence ou l'absence du gène, de l'enzyme et de la voie métabolique. L'absence de la section '*Gene'* dépend de l'annotation du gène codant si elle est complète ou non; on retrouve ce cas (l'absence de la section '*Gene'*) dans la base de données de l'espèce *Pseudomonas chlororaphis*. L'absence de la section '*Enzyme'* dépend de l'absence de l'activité enzymatique de la protéine décrite dans l'entrée. La même chose pour la section '*Pathway'* qu'on peut la retrouver ou non sur une entrée de ***PseudomonasDW*** selon la participation ou non de la protéine dans des voies métaboliques.

# 5 OUTILS BIOINFORMATIQUES DE PSEUDOMONASDW

Nous avons vu précédemment dans le chapitre I de ce manuscrit que les données biologiques continuent de croître de manière exponentielle, tant en nombre qu'en types. Qu'elles soient, des séquences, des profils d'expression, des polymorphismes ou des entrées bibliographiques, il a été nécessaire de développer des outils pour interroger ou recouper ces données et permettre aux utilisateurs de comparer leurs propres données à l'existant.

Ces outils doivent donc être :

- Facilement accédés, c'est à dire librement accessibles via Internet ;
- Didactiques, c'est à dire faciles à prendre en main, voire, mieux encore, intuitifs ;
- Exhaustifs, c'est à dire qu'à partir d'une information trouvée, ils doivent permettre de parcourir l'ensemble des liens rattachés à celle-ci afin d'éviter à l'utilisateur d'être obligé de jongler avec différentes sources d'informations.

Deux grands types d'outils sont à présent disponibles pour la communauté des biologistes, les navigateurs de banques de données[91] et les navigateurs génomiques[92]. Les premiers sont dédiés à l'interrogation des banques et bases de données, tandis que les deuxièmes, sont comme leur nom l'indique, dédiés au parcours de génomes complets et à la visualisation des annotations associées. Cette classification est toutefois quelque peu schématique puisque certains outils intègrent l'ensemble des fonctionnalités : bases de données, outils d'interrogation et outils de navigation sur le génome.

C'est pourquoi, une telle base de données comme ***PseudomonasDW***, a l'obligation aujourd'hui, d'intégrer dans son application web, différents outils bioinformatiques destinés à faciliter l'exploitation et l'analyse de ses données, notamment, un navigateur génomique qu'est devenu indispensable pour une base de donnée génomique. Pour combler ce manque, nous nous sommes chargés d'accomplir une tâche essentielle, d'abord choisir et intégrer un navigateur génomique pour ***PseudomonasDW***, et ensuite intégrer un autre outil d'alignement de séquences qui permet aux utilisateurs de trouver les régions similaires entre deux ou plusieurs séquences nucléotidiques ou peptidiques de différentes espèces stockées dans ***PseudomonasDW***.

## 5.1 Navigateur génomique pour PseudomonasDW (GBrowse)

Le choix d'un navigateur génomique, pour ***PseudomonasDW*** est une tâche qui n'est pas facile ni évidente, du fait, que les différents navigateurs génomiques présentent plusieurs points forts et plusieurs faiblesses.

[91]DataBank browsers
[92]Genome browsers

Par exemple, l'un des plus populaires navigateurs génomiques qui est *Ensembl*, présente la meilleure application pour la génomique comparative, mais d'autre part, un autre navigateur génomique populaire qui est *Gbrowse*[93] offre une meilleure flexibilité avec beaucoup d'options supplémentaires et de PlugIns, en addition d'une large communauté de développeurs ainsi que le grand nombre de bases de données génomiques de référence et qui ont une bonne réputation, mais son application pour la génomique comparative n'est pas aussi riche que *Ensembl*.

Par conséquent, la détermination du navigateur génomique qui convient le mieux aux besoins des chercheurs et l'ensemble de la communauté scientifique qui s'intéresse à Pseudomonas sp, est une étape clé dans cette thèse, et une tâche qui requiert un examen attentif.

Ainsi, plusieurs raisons ont contribué à notre choix final, de *Gbrowse* comme navigateur génomique, pour ***PseudomonasDW***:

- *Ensembl*, est toute une application libre de droit d'auteur sur son code source, qui pourra techniquement être adaptés à ***PseudomonasDW***, et fait tout le nécessaire dans un navigateur génomique. Mais il est de moins en moins utilisé et son communauté de développeurs n'est pas aussi large que celle de *Gbrowse* ce qui rend son développement moins actif, sa mise-à-jour moins fréquente et la découverte et la résolution de bugs plus difficile.
- L'intégration d'un navigateur génomique bien connu et plus utilisé, présente des avantages considérables. A court terme, il est préférable et bien recommandé que les utilisateurs potentiels de ***PseudomonasDW*** soient familiarisés avec le fonctionnement du navigateur génomique qui serait mis à leur disposition dans le site Web. Or la plupart des bases et banques de données génomiques existantes et qui s'intéressent à Pseudomonas sp emploie *Gbrowse* comme navigateur génomique, c'est à dire qu'il est l'outil avec lequel les futurs utilisateurs potentiels ont l'habitude de travailler, par conséquent, ils le trouveront plus aisé à manipuler.
- Les caractéristiques les plus désirées et les plus demandée dans un navigateur génomique sont la facilité d'utilisation, la visualisation claire et intuitive des génomes en plus de la rapidité qui est indispensable.

Plusieurs sondages réalisés à ce propos montrent que les utilisateurs des navigateurs génomiques, en général, ne considèrent pas *Ensembl* facile et intuitive en comparaison aux autres navigateurs (Sen, et al., 2010).

[93] http://gmod.org/wiki/GBrowse

### 5.1.1 GBrowse : Vue générale

*GBrowse* est une partie du projet GMOD (*Generic Modele Organisme Database project*) qui correspond à une collection de logiciels open source pour créer et gérer des bases de données biologiques à l'échelle du génome. Le projet GMOD est soutenu par un accord spécifique de coopération entre le Service pour la recherche agricole de l'USDA, et par des subventions des NIH co-financées par le *National Human Genome Research Institut* et *l'Institut national des sciences médicales générales*. Ce projet est sous licence GNU* General Public License (ou GPL)

*GBrowse* a été désigné pour la visualisation des génomes, il affiche une représentation graphique d'une section d'un génome, ainsi que les positions des gènes en plus d'autres éléments fonctionnels. *GBrowse* peut être configuré pour afficher les données qualitatives comme la structure d'un gène ou quantitative comme les degrés d'expression des puces à ADN. *GBrowse* propose les fonctionnalités suivantes :

- vue globale et vue détaillée du génome,
- défilement, zoom et centrage,
- utilisation de représentations graphiques (ou glyphes) préfabriquées, ou bien personnalisées,
- joindre une URL arbitraire à une annotation,
- ordre et apparence des pistes personnalisables par l'administrateur et l'utilisateur final,
- recherche par ID annotation, nom ou commentaire,
- connectivité à différentes bases de données, telles que BioSQL[94] et Chado[95],
- support multi-langues,
- prise en charge des annotations à partir du format GFF[96],
- persistance des paramètres de session à session,
- plug-in* d'architecture personnalisable (par exemple exécuter BLAST, importer de nombreux formats, trouver des oligonucléotides*, concevoir des amorces, créer des cartes de restriction, éditer des fonctions).

### 5.1.2 Installation de GBrowse

Le serveur qui héberge ***PseudomonasDW*** est sous la plateforme Linux, sur ce fait nous avons choisi d'utiliser un shell CPAN (réseau complet d'archives Perl) qui facilite l'installation des prérequis fondamentales pour le fonctionnement de *GBrowse*. Nous avons eu besoin d'installer :

[94] http://www.biosql.org/wiki/Main_Page
[95] http://gmod.org/wiki/Chado_-_Getting_Started
[96] http://gmod.org/wiki/GFF

- Apache Web Server[97]
- Perl 5[98]
- Les modules de Perl suivants:
  - GCI
  - GD
  - DBI
  - DBD ::mysql
  - Digest ::MD5
  - Text ::shellwords
- Bioperl[99]

Il existe plusieurs méthodes pour installer *Gbrowse*, premièrement nous avons choisi d'installer *Gbrowse2*, nous avons utilisé la commande *apt-get* qui nous a permis une installation automatique de *GBrowse*.

```
admin@admin:~$ sudo apt-get install gbrowse gbrowse-calign
gbrowse-data
```

La façon optimale et recommandée pour l'intégration de *GBrowse* est de mettre les données d'intérêts dans des bases de données. *GBrowse* supporte plusieurs systèmes de gestion de bases de données grâce aux nombreux adaptateurs dont il dispose, chacun avec sa vitesse, ces avantages, ses limites et ses types de formats qu'il supporte. A cette étape d'installation nous étions encore confrontés à faire un choix parmi la multitude des adaptateurs disponibles. Côté format de fichiers, il est mentionné souvent dans la littérature que le format optimal pour stocker les données génomiques est le format GFF3, le SGBD le plus adéquat étant MySQL, d'abord parce qu'il est le plus utilisé et ensuite parce qu'il est le premier implémenté dans *GBrowse*, donc il a acquis plus d'expériences et d'améliorations au fil des années. Nous avons choisi l'adaptateur **Bio::DB: SeqFeature::Store** pour assurer la communication entre *GBrowse* et les bases de données MySQL. L'adaptateur **Bio::DB: SeqFeature::Store** est le plus adapté à fonctionner avec GFF3 et MySQL, il est d'ailleurs le plus récent des adaptateurs et le plus recommandé.

### 5.1.3 Création et peuplement des bases de données MySQL

Avant la création et le peuplement des bases de données, l'obtention des données est une étape qui nécessite une étude minutieuse. Les données génomiques fournies par ***PseudomonasDW*** concernent seulement les gènes codant pour des protéines (puisque chaque entrée de ***PseudomonasDW*** décrit une protéine et les différentes données relatives à cette protéine) et manquent aux autres loci génomiques. Notons dans ce

[97] http://httpd.apache.org/
[98] http://dev.perl.org/perl5/
[99] http://www.bioperl.org/wiki/Main_Page

contexte que les données génomiques utilisées par ***PseudomonasDW*** proviennent de la banque de données GenBank, pour cela nous avons choisi d'utiliser et d'adapter (selon nos besoins) les fichiers GFF3 fournies par GenBank pour combler le manque de nos fichiers GFF3.

La Figure 45 explique les différentes étapes de création et de configuration de bases de données MySQL. La première étape après l'adaptation des fichiers GFF3 de GenBank était la création de **34** bases de données pour **29** éspèces de Pseudomonas intégrées dans ***PseudomonasDW*** (**29** bases de données pour les chromosomes et **5** bases de données pour les plasmides). La deuxième étape était le peuplement de chaque base de données MySQL par le contenu du fichier GFF3 correspondant ; cette étape a était réalisée par l'exécution du module de Bioperl '*bp_seqfeature_load.pl*' en utilisant le code suivant:

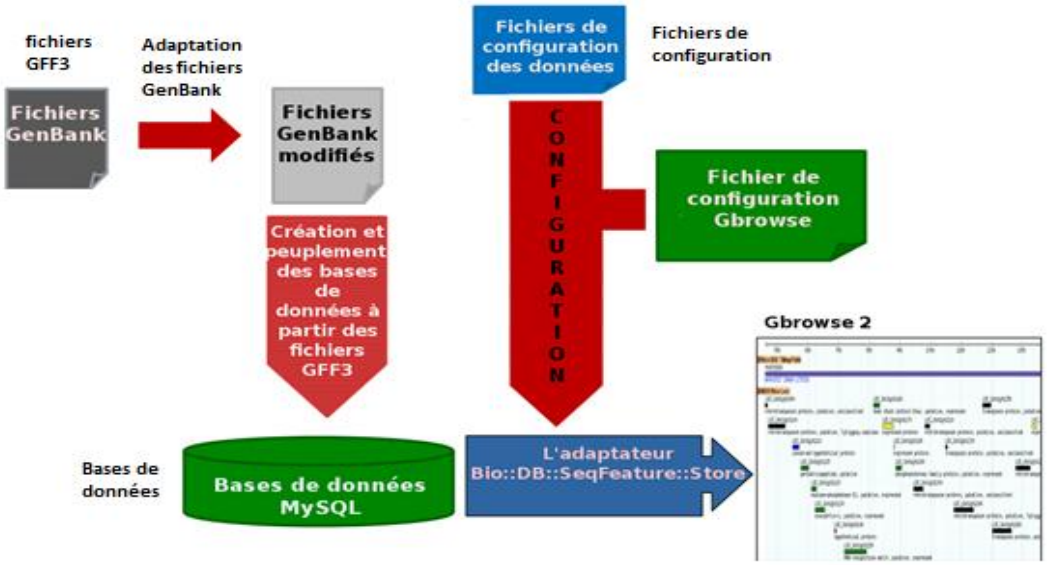


**Figure 45. Les différentes étapes de création de bases de données de GBrowse**

```
admin@admin:~$ sudo bp_seqfeature_load.pl -c --dsn
"dbi:mysql:DB_Name" --user "root" --password "*****"
/var/lib/gbrowse/databases/file.gff3
```

La dernière étape était la configuration des bases de données MySQL pour qu'elles soient lisibles et accessibles par l'outil *GBrowse*. Cette étape a était réalisée via la création de fichier de configuration pour chaque base de données. Le fichier de configuration garde la forme générale du fichier '*GBrowse.conf*' qui se crée automatiquement lors de l'installation de *GBrowse* et qui contient les directives qui indiquent à l'outil les instructions d'options qui

s'appliquent sur l'ensemble des bases de données. Cependant, nous avons édité le paramètre **db_adaptor = Bio::DB::SeqFeature::Store** dans chaque fichier de configuration pour faciliter la communication entre *GBrowse* et les bases de données. Ainsi, nous avons introduit quelques modifications concernant les paramètres d'affichage pour données une lisibilité à l'image de *GBrowse* résultante.

Afin d'adapter le fonctionnement de ***PseudomonasDW*** avec l'intégration de *GBrowse*, nous avons ajouté, pour chaque section Gene de chaque entrée de ***PseudomonasDW***, un onglet intitulé *Gbrowse View* qui se charge d'afficher l'image du gène correspondant à l'entrée (**Figure 46**). Pour une recherche plus exhaustive, l'utilisateur peut naviguer vers l'outil *GBrowse* intégré au niveau de ***PseudomonasDW*** en cliquant seulement sur l'image résultante.

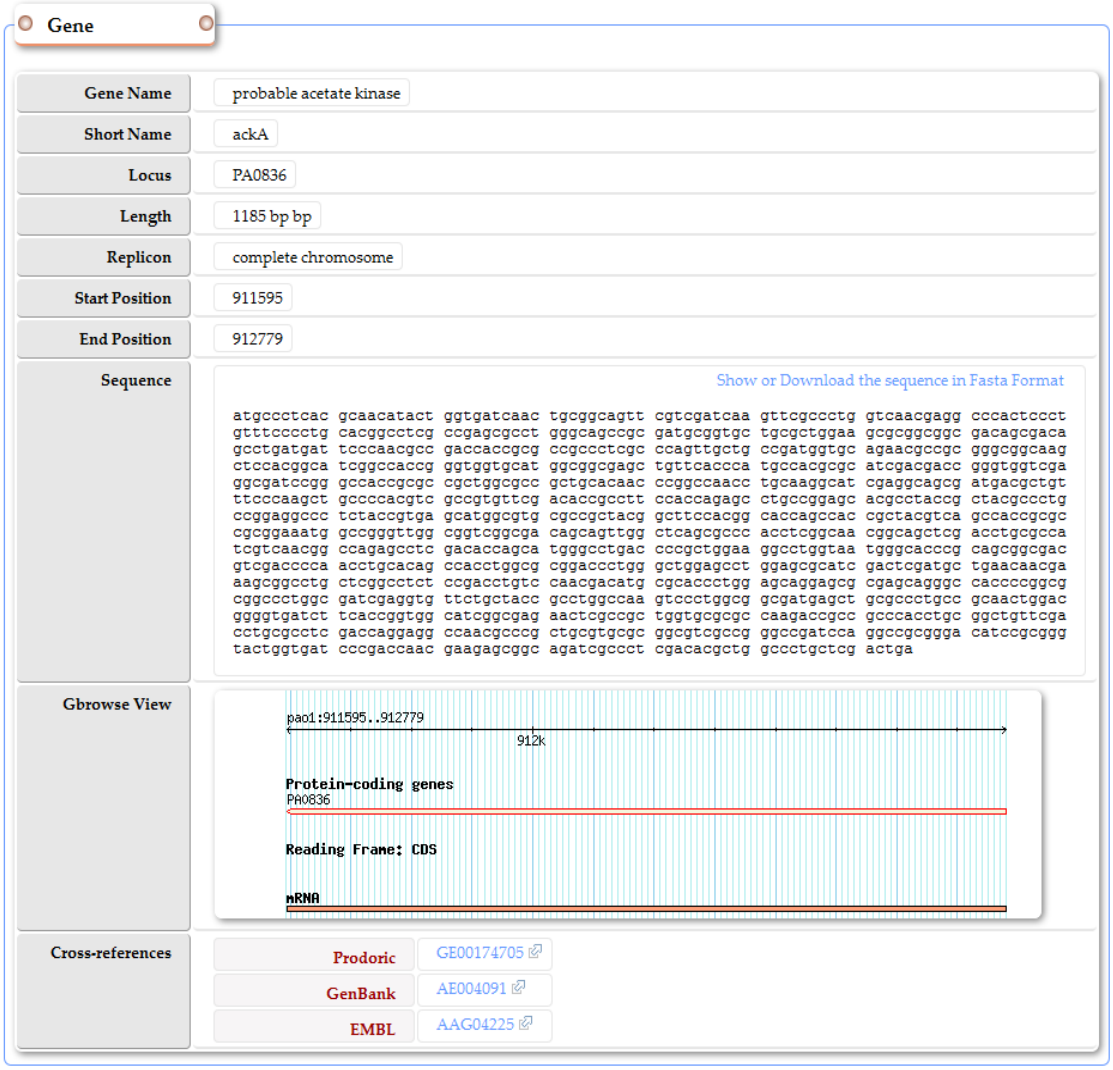


**Figure 46. L'image de GBrowse intégrée dans la section 'Gene' de l'entrée 'PAE00011'**

## 5.2 Intégration de l'outil Blast dans PseudomonasDW

### 5.2.1 Blast : Vue générale

*Blast* est un programme permettant de réaliser un alignement local entre deux séquences (nucléiques ou protéiques). Sa rapidité permet d'effectuer des comparaisons entre une séquence donnée, dite requête, et un ensemble de séquences. *Blast* est fourni sous la forme d'un "package", composé des programmes suivants:

- **blastn** blast nucléique :
  Pour comparer une séquence requête nucléique à une banque de séquences nucléiques
- **blastp** blast protéique :
  Pour comparer une séquence requête protéique à une banque de séquences protéiques
- **blastx** blast nucléique vs protéique :
  Pour comparer une séquence requête nucléique à une banque de séquences protéiques
- **tblastn** blast protéique vs nucléique :
  Pour comparer une séquence requête protéique à une banque de séquences nucléiques
- **tblastx** blast nucléique vs nucléique en passant par un alignement protéique :
  Pour comparer une séquence requête nucléique à une banque de séquences nucléiques en alignant les séquences protéiques induites par les séquences nucléiques.

L'intégration de *Blast* dans ***PseudomonasDW*** n'était pas une tâche laborieuse comme celle du *GBrowse*. La première étape dans l'intégration de *Blast*, après avoir téléchargé son package, était la création des bases de données utilisable par le *Blast* : une base de données pour chaque espèce intégrée dans ***PseudomonasDW***. Le programme '*makeblastdb'* fourni dans le package BLAST, permet de créer automatiquement une telle base de données, à partir de nos séquences stockées au format FASTA.

Cependant, l'objectif de cette partie de travail n'était pas une installation de Blast, mais son intégration au sein de ***PseudomonasDW*** pour permettre aux utilisateurs de l'entrepôt de données de faire un blast de leurs séquences contre les différentes bases de données proposées par ***PseudomonasDW***. Ainsi pour atteindre cet objectif, nous avons développé une application Web capable de soumettre les requêtes des utilisateurs à *Blast*. Cette application est installé sur le serveur de ***PseudomonasDW*** pour recevoir la réponse et de le transmettre à son tour à l'utilisateur dans un navigateur Web.

### 5.2.2 La fonctionnalité du Blast

L'utilisateur de ***PseudomonasDW***, désirant comparer sa propre séquence avec les séquences contenues dans les bases de données de ***PseudomonasDW***, peut accéder à la page réservée à Blast via le menu gauche de la page d'accueil du site Web de ***PseudomonasDW***. La Figure 47 montre une capture d'écran de la page Web du Blast dans ***PseudomonasDW***.

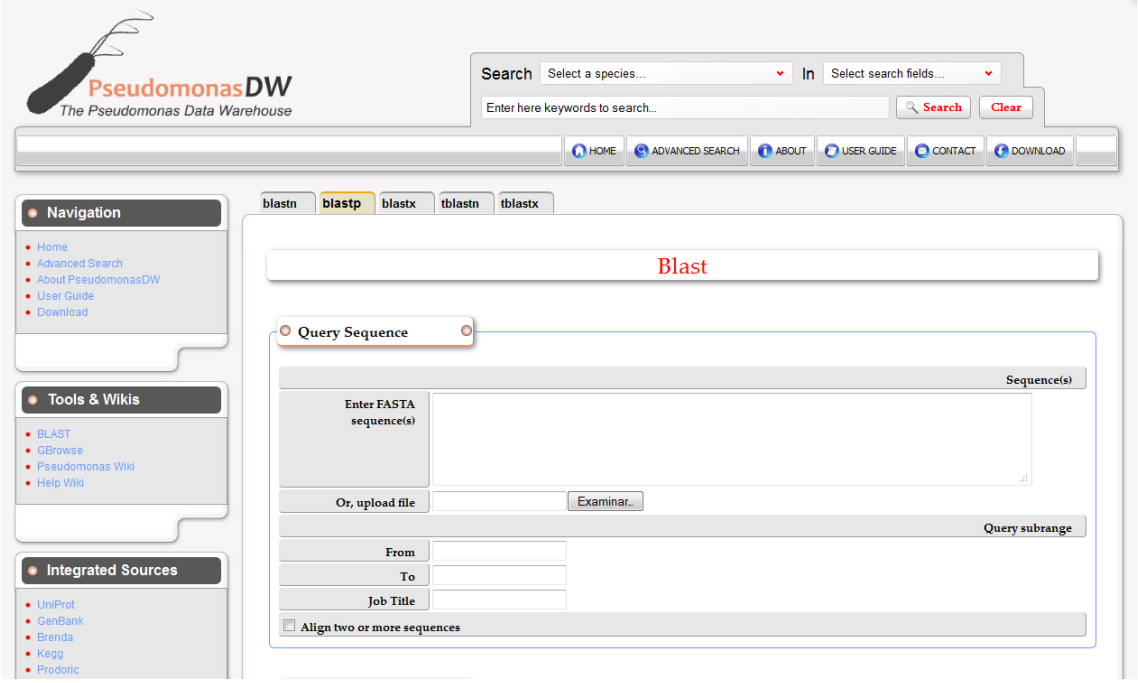


**Figure 47. Capture d'écran montrant la page Web du Blast dans PseudomonasDW**

La page Web du Blast fournit par le site de ***PseudomonasDW*** offre à l'utilisateur la possibilité de "blaster" ses séquences contre :

Les différentes bases de données de ***PseudmonasDW*** par la soumission des séquences (nucléiques ou peptidiques) ou par le chargement d'un fichier texte contenant les séquences à aligner en format FASTA. L'utilisateur peut aligner contre une seule base de données comme il peut aligner contre toutes les bases de données de ***PseudomonasDW*** par le choix de l'option « All Databases » (Figure 48). L'utilisateur a la possibilité aussi de définir la partie de la séquence qu'il souhaite aligner en déterminant les coordonnées de ses extrémités.

Un ensemble de séquences de son choix en faisant appel à un deuxième formulaire de soumission en cochant la case « Align two or more sequences » (Figure 49). Cette

option offre la possibilité d'aligner deux ensembles de séquences indépendamment des bases de données stockées au niveau de ***PseudomonasDW***.

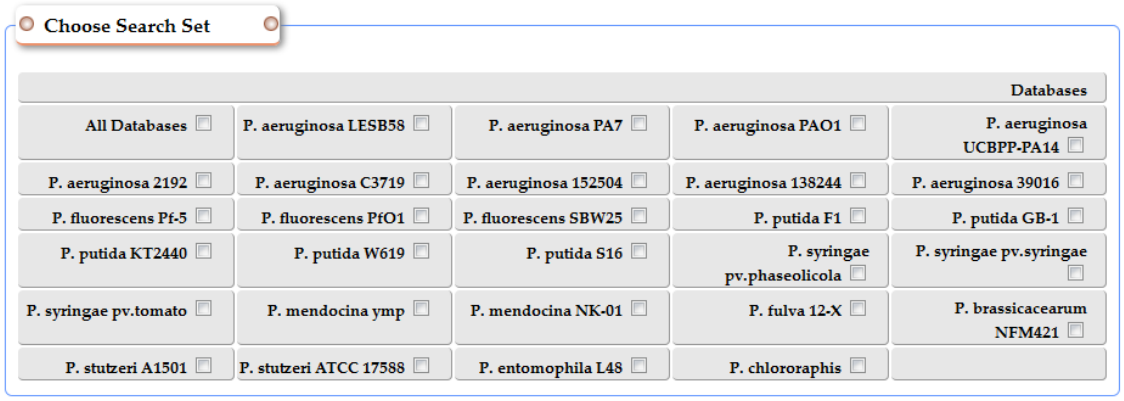


**Figure 48. Une capture d'écran montrant les différentes bases de données parmi lesquelles l'utilisateur peut choisir.**

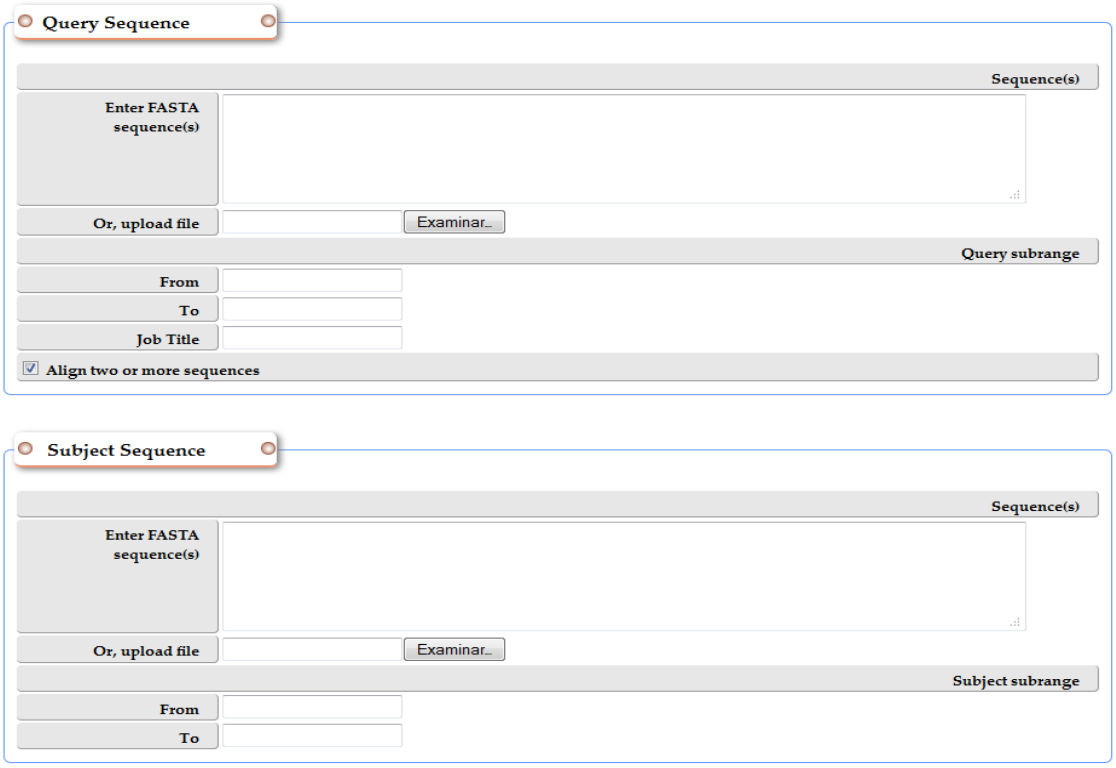


**Figure 49. Une capture d'écran montrant la possibilité d'aligner deux ensembles de séquences indépendamment des bases de données de PseudomonasDW**

Pour le traitement de la requête de l'utilisateur, nous avons développé une servlet Java '*RunBlast'* qui se charge de prendre les données envoyées via la requête, les analyser et en extraire les paramètres nécessaires tels que le type de séquence (protéique/nucléique) et le sous-programme utilisé (blastn, blastp, blastx…) et enfin, les attribuer comme valeurs d'attributs d'un objet instancié d'une classe Java '*BlastSeq.java'* que nous avons aussi développé. Cette classe possède une méthode qui nous permet de générer dynamiquement une commande à envoyer au sous-programme choisi de *Blast* et d'en recevoir la réponse qui sera retournée à l'utilisateur via son navigateur Web.

Le résultat affiché, pour l'utilisateur, est composé de trois sections : la section '*General Information'* qui offre des informations sur la requête envoyée en déterminant le programme de *Blast* choisi, le nom de la base de données à laquelle appartient la séquence soumit, une petite définition de la séquence en déterminant le nom du gène, le nom de la protéine, l'espèce et la langueur de la séquence. La deuxième partie '*Description*' décrive les différentes séquences alignées avec la séquence en question en déterminant leur numéro d'accession dans ***PseudomonasDW***, leurs bases de données, les noms du gène et de protéine et les scores de similarités. La dernière section '*Alignment'* montre les alignements obtenus en déterminant tous les paramètres de l'alignement (le score de l'alignement, le pourcentage d'identité et le pourcentage des gaps) et en donnant une image générale de l'alignement obtenu. La (Figure50) montre les trois sections du résultat du Blast et un exemple d'alignement.

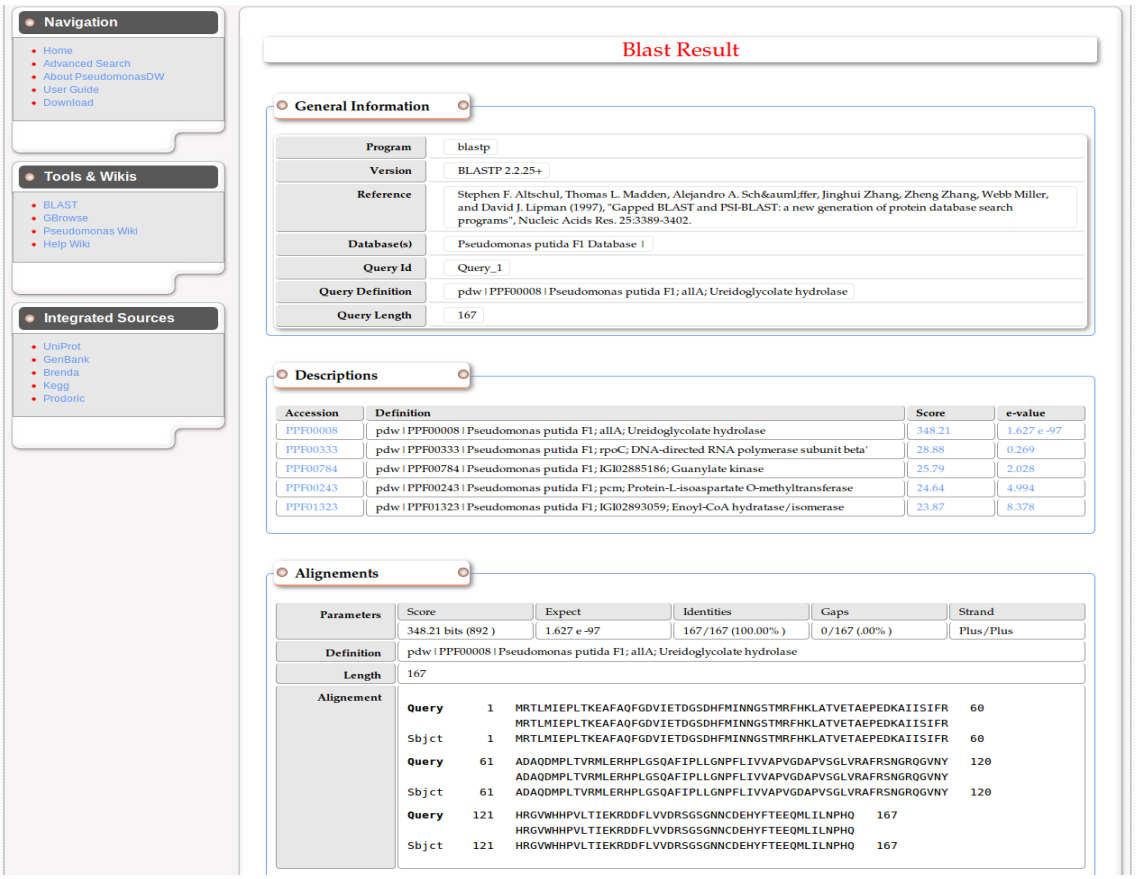


**Figure50. Exemple de résultat de Blast**

# 6 PDWiki

Pour rendre l'entrepôt de données ***PseudomonasDW*** plus informatif, nous avons développé un Wiki scientifique nommé **PDWiki**. L'idée principale derrière **PDWiki** est de donner à la communauté scientifique de Pseudomonas de trouver, éditer et ajouter des informations relatives aux organismes, les gènes, les protéines, les enzymes et les voies métaboliques intégrés dans ***PseudomonasDW***. Ces informations pourraient être d'intérêts différents comme la microbiologie, la biologie médicale et la biologie évolutive.

Dans cette section de ce quatrième chapitre, nous donnons une vue générale sur les Wiki biologiques en déterminant leurs intérêt dans le domaine biologique et aussi nous introduisons **PDWiki** en décrivant ses composants, sa méthode d'implémentation et sa manière d'accès.

## 6.1 Généralité sur les Wikis biologiques

Le succès des projets communautaires tels que Wikipedia[100] a récemment suscité un débat sur l'application des wikis dans les sciences de la vie. Un wiki est un outil basé sur le Web sert à assurer la conservation et l'édition d'un ensemble de pages Web. Il fournit un cadre simple pour capturer et partager des données, générée par tout utilisateur disposant d'un navigateur Web et les autorisations appropriées pour modifier le contenu du wiki. Il est maintenant clair que les systèmes de wiki offrent une variété d'avantages pour la gestion des données et des informations biologiques. Certains des objectifs spécifiques de wikis biologiques (bio-wikis) comprennent:

- Le développement collaboratif et le partage des connaissances
- L'annotation collaborative de contenus de bases de données
- La création collaborative de contenus de bases de données

**Le développement collaboratif et le partage de la documentation et des connaissances** permet aux collectivités de promouvoir, d'exploiter, de discuter un consensus sur l'information, des procédures, des données, des nouvelles expériences, des nouvelles, et d'autres informations variées. Cet objectif est motivé par la prise de conscience que l'expertise et les intérêts précieux, sur des sujets spéciaux, sont généralement distribués, et sont rarement concentrées dans un site ou d'un groupe de recherche unique. L'objectif est la mise en œuvre des recueils de haute qualité sur des sujets biologiques spécialisés.

**L'annotation collaborative de bases de données biologiques** s'appuie sur le fait que la curation précise et étendue d'un volume croissant de données est extrêmement coûteuse et chronophage. L'objectif est d'améliorer et d'étendre la curation des bases de données delà de ce qui est possible avec un petit groupe de curation. Elle permet aux utilisateurs d'apporter leur expertise, leurs expériences, leurs observations et leurs résultats, indépendamment de l'organisation de la base de données. Les utilisateurs peuvent contrôler cette curation étendue, corriger et mettre à jour des archives dans les meilleurs délais. Bien que le contenu des bases de données soit annoté d'une manière collaborative, les bases de données elles-mêmes restent inchangées.

**La création collaborative de base de données** capture la structure émergente dans les domaines qui se développent rapidement. Ces bases de données sont des indices de données biologiques pertinentes qui se dégagent de communautés ciblées et rapidement développées. Elles forment un pis-aller entre la discussion non structurée dans les forums et sur les listes de diffusion et les bases de données «matures» qui émergent par la suite.

[100] http://www.wikipedia.org/

## 6.2 PDWiki: Infrastructure et contenue

PDWiki est implémenté en utilisant MediaWiki[101] : une application libre de logiciel wiki basée sur le Web et écrite en PHP. Ce logiciel est optimisé pour développer efficacement et correctement des projets de n'importe quelle taille. Il est fortement personnalisé avec des extensions et des paramètres[102] de configurations multiples disponibles pour l'activation de différentes fonctionnalités pour être ajoutées ou modifiées[103]. Plusieurs robots[104] automatisés ou semi-automatisés ont été développés pour aider l'édition des sites de MediaWiki.

MediaWiki nous a permis de créer un ensemble très large de pages en utilisant de nombreuses fonctionnalités d'annotations intégrées. Ces pages ont été créées au moyen des robots que nous avons implémenté par le Framework[105] Java Bot Wiki : une bibliothèque pour maintenir les wikis basés sur MediaWiki, il prend en charge l'API de MediaWiki et fournit des méthodes pour se connecter, modifier et lire des collections. Le principal robot que nous avons créé est celui qui nous a permis de parcourir les entrées des bases de donnes de ***PseudomonasDW*** et de créer une page de wiki pour chaque entrée de l'entrepôt. Ce rebot est composé de trois classes Java : '*DatabaseParser*' '*Template'* et '*Bot'*. La classe '*DatabaseParser'*, en utilisant le JAXP, offre des méthodes pour parcourir les entrées de ***PseudomonasDW*** et extraire les informations nécessaire pour construire la classe '*Template'* , qui à son tour, construit la structure de base des pages de PDWiki. La classe '*Bot'* est la classe principale du robot ; elle se connecte à PDWiki et transforme la structure générée par la classe '*Template*' en une page réelle de PDWiki. La classe 'Bot' interagie avec PDWiki comme s'il est un éditeur humain. Elle crée une page vide de PDWiki dans laquelle elle reflète le contenue du résultat de la classe '*Template'*.

PDWiki dispose de deux types de pages: des pages liées aux entrées de ***PseudomonasDW*** '*PDWEPS'* (Figure 51) et des pages génériques '*GPDWiPs'*. Le premier type vise à annoter les entrées de ***PseudomonasDW*** en tenant des informations supplémentaires non disponibles dans les bases de données de ***PseudomonasDW***. Pour chaque entrée de ***PseudomonasDW*** il y a une page '*PDWEP'* : ce qui donne un total de plus de **170.000** pages de *PDWEP*. Chacune de ces page est divisée en, mais n'est pas limitée à, sept sections principales: '*General Information'*, '*Gene'*, '*Protein'*, '*Enzyme'*, '*Pathway'* et '*References'*. Les utilisateurs ont la possibilité d'étendre ces sections en créant d'autres plus.

La section des '*General Information'* contient des informations de base sur l'entrée correspontante dans ***PseudomonasDW***. Cela inclut le numéro d'accession de l'entrée dans ***PseudomonasDW***, le nom du gène, le nom de protéines, la fonction des protéines, et le

[101] http://www.mediawiki.org/wiki/MediaWiki
[102] http://www.mediawiki.org/wiki/Category:MediaWiki_configuration_settings
[103] http://www.mediawiki.org/wiki/Extension_Matrix
[104] http://en.wikipedia.org/wiki/Wikipedia:Bots
[105] http://jwbf.sourceforge.net/

nom de l'organisme. Le numéro d'accession est lié à son entrée associée dans ***PseudomonasDW*** via un lien hypertexte. La section '*General Information*' n'est pas modifiable par l'utilisateur et les données sont obtenues directement à partir ***PseudmonasDW***.

La section '*Organism*' détient le nom de l'espèce de la page '*PDWEP*' à laquelle elle appartient; cette section peut également contenir des informations décrivant cette espèce. Chaque espèce de Pseudomonas intégrées dans ***PseudomonasDW*** dispose d'une page spécifique (une page *GPDWiP*) dans PDWiki qui peut contenir des informations supplémentaires sur l'espèce. La page '*GPDWiP*' est : (1) accessible en cliquant sur le nom de l'espèce indiqué dans la section '*Organism*' de la page '*PDWEP*' et (2) structurée selon, au moins, six sections: '*Taxonomy*', '*Description*', '*Characteristics*', '*Genome*', '*Statistics*', et '*References*' . La section '*Statistics*' informe les utilisateurs sur le nombre d'entrées concernant chaque espèce intégrée dans ***PseudomonasDW*** et fournit un lien pour accéder à une page '*GPDWiP*' qui liste toutes ces entrées. En cliquant sur un élément de la liste, l'utilisateur est conduit vers une page '*PDWEP*' qui annote l'entrée de ***PseudomonasDW***.

Les sections '*Gene*', '*Protein*', '*Enzymes*', et '*Pathways*' sont toutes modifiables. Les utilisateurs peuvent modifier ou mettre à jour les informations sur le gène présenté par l'entrée de ***PseudomonasDW*** dans la section '*Gene*' tandis que, dans la section '*Protein*', ils peuvent modifier ou mettre à jour les informations relatives au produit du gène. Ces informations peuvent inclure des maladies associées à des anomalies de la protéine, les interactions avec autres protéines, des informations issues des expériences de spectrométrie de masse, des propriétés biophysiques et physico-chimiques ... etc. D'autre part, les sections '*Enzymes*' et '*Pathways*' sont réservées, respectivement, pour les enzymes et les voies métaboliques liées à la protéine annotée dans la section '*Protein*'. Alors que les utilisateurs peuvent modifier ou ajouter dans la section '*Enzymes*', par exemple, les informations des réactions catalysées par l'enzyme, les substances non protéiques nécessaires pour les activités enzymatiques; le mécanisme réglementaire de l'enzyme, il est possible de modifier les voies métaboliques associées en donnant une description générale ou en éditant des informations supplémentaires, sur leurs listes des métabolites ou leurs différents composants, dans la section '*Pathways*'.

Enfin, la section '*References*' contient des citations de la littérature qui sont les sources d'information utilisées pour modifier le '*PDWEP*'. Chaque référence est numérotée et contient plusieurs sous-sections permettant une description précise d'une citation donnée.

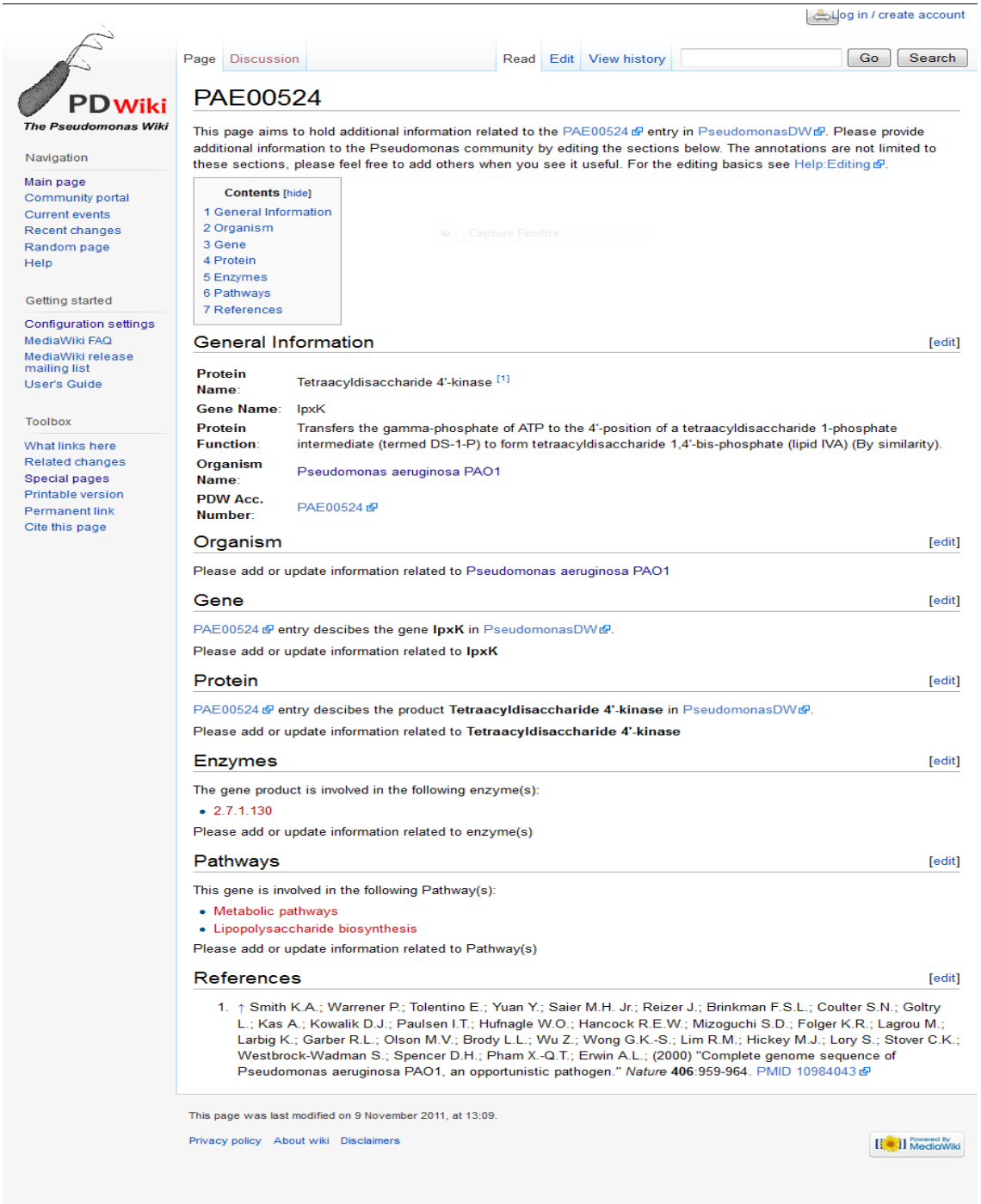


**Figure 51. Un exemple d'une page PDWEP. Elle concerne la page de PDWiki créée pour enrichir et annoter l'entrée PAE00524 de PseudomonasDW**

'*GPDWiPs*' sont toutes les pages de PDWiki autres que '*PDWEPs*' (Figure 52). Ils contiennent des informations génériques relatives aux espèces de Pseudomonas intégrées dans ***PseudomonasDW*** ou un de leurs composés cellulaires. Des exemples de '*GPDWiPs*'

pourrait être une espèce ou une page souche (ex ; la page de Pseudomonas aeruginosa ou la page de Pseudomonas aeruginosa PAO1), une page reliée à une enzyme (page protéase alcaline), une page d'une toxine intracellulaire (la page ExoA, la page ExoS), une page des gènes reliée à une espèce (la page Pseudomonas aeruginosa PAO1 genes) et ainsi de suite.

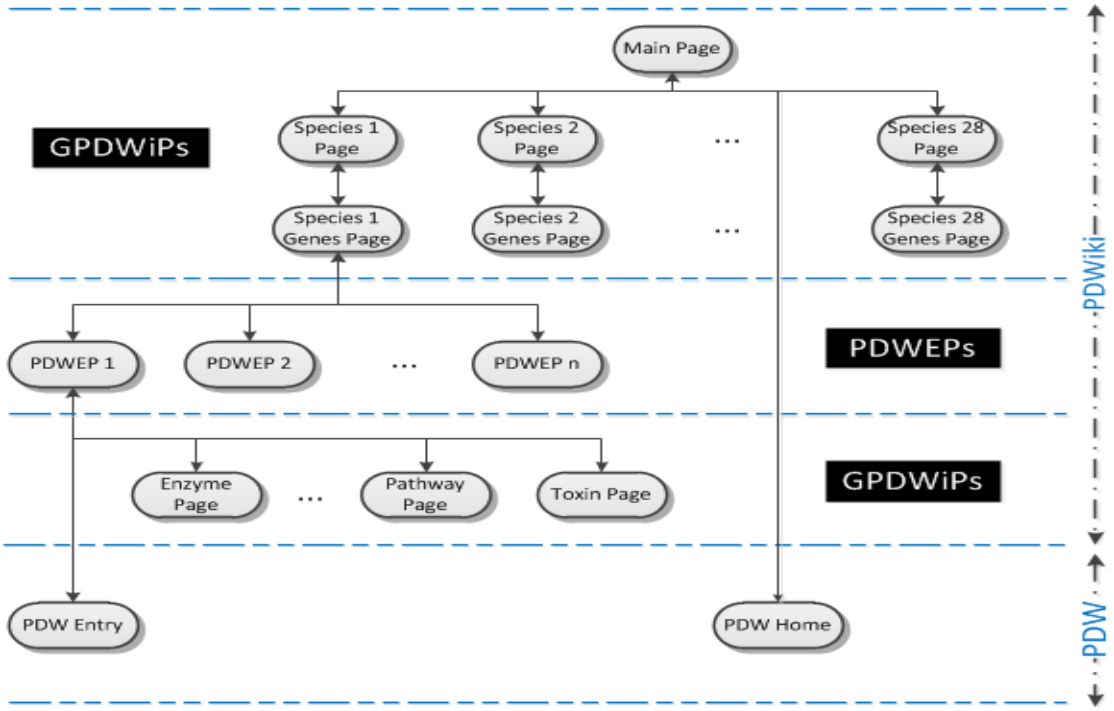


**Figure 52. Un schéma déscriptif de la structure de PDWiki. Il montre la structure de base de PDWiki et les relations entre ses pages et PseudomonasDW (PDW)**

'*GPDWiPs*' ont été créés pour tenir plus d'annotations. De point de vue modélisation, ces pages pourraient être considérés, dans certains cas, comme une généralisation de certains '*PDWEPs*'; on peut citer le cas les pages des gènes des espèces qui contiennent une liste alphabétique ordonnée de tous les gènes d'une espèce, de Pseudomonas et à partir de cette page, il est possible d'aller à un '*PDWEP*' spécifique en cliquant sur le nom d'un gène. D'autres cas, des pages '*GPDWiPs*' sont des spécialisations de certains pages de '*PDWEPs*'. C'est le cas par exemple d'une information tenue par une page '*GPDWiP*' sur une voie métabolique apparaissant dans une page '*PDWEP*'.

## 6.3 Comment naviguer dans PDWiki

Pour les utilisateurs qui ne sont pas familiarisés avec les wikis basés sur MediaWiki, la recherche est le processus le plus simple et plus puissant qui leurs permet de trouver des pages spécifiques dans PDWiki. Une barre de recherche est située sur le côté supérieur

gauche de chaque page constituée par un champ de recherche, un bouton '*GO'* qui apparaît sur toutes les pages de PDWiki à côté d'un bouton '*Search'*. La fonction du bouton '*GO'* est de naviguer directement à la page dont son nom est le texte édité dans le champ de recherche, alors que la fonction de bouton '*Search'* est la recherche du texte dans toutes les pages de PDWiki. Ainsi, l'utilisateur peut commencer à trouver l'information souhaitée au sein de PDWiki en utilisant le formulaire de recherche.

Les utilisateurs de PDWiki peuvent également obtenir des informations sur chaque espèce ou souche dans PDWiki en suivant les liens sur la page d'accueil qui conduisent à une page '*GPDWiP'*. En outre, il y a une sorte de navigation bidirectionnelle entre ***PseudomonasDW*** et PDWiki: à partir d'une entrée de ***PseudomonasDW***, il est possible d'aller vers la page '*PDWEP'* correspondante dans PDWiki et vice-versa.

Toutes les pages de PDWiki sont accessibles au public. En revanche, il est obligatoire de s'enregistrer pour éditer ou modifier des pages de PDWiki. C'est une démarche simple et rapide, il suffit que l'utilisateur crée un compte utilisateur personnel. Cette action a plusieurs avantages certains d'entre eux sont:

- Les utilisateurs seront capables de reconnaître les uns des autres par '*usermane'* quand quelqu'un fait des modifications au niveau des pages de PDWiki.
- L'utilisateur aura sa propre page où il peut écrire des informations sur lui-même, et une page de discussion dont il peut l'utiliser pour communiquer avec d'autres utilisateurs.
- L'utilisateur sera capable de garder une trace des modifications apportées aux pages qui lui intéresse en utilisant la fonctionnalité '*watchlist'*[106].

# 7 DISCUSSION

Certaines espèces de Pseudomonas sont désormais considérées comme des organismes modèles et ont été largement étudiées en raison de leur résistance antimicrobienne (Rehm, 2009), diverse capacités métaboliques, et sa capacité de causer des infections graves. Plusieurs systèmes de haute qualité pour la recherche de données biologiques de Pseudomonas et leurs annotations ont été cités dans l'introduction de ce chapitre. Dans cette section, nous présentons une brève comparaison entre ***PseudomonasDW*** et la base de données « Pseudomonas Genome database » (Winsor, et al., 2009), qui est l'une des bases de données célèbres intéressées par l'annotation de Pseudomonas et la plus similaire à la philosophie de ***PseudomonasDW***. Cette base de données se concentre sur l'annotation du génome de Pseudomonas aeruginosa PAO1 et fournit des informations les plus pertinentes pour la recherche de Pseudomonas aeruginosa. Pour d'autres souches de

[106] http://www.mediawiki.org/wiki/Manual:Watchlist

Pseudomonas, elle donne un grand ensemble d'informations, mais reste modeste en comparant à Pseudomonas aeruginosa PAO1. En revanche aux bases de données ***PseudomonsDW*** qui se concentrent sur les protéines Pseudomonas, la base de données « Pseudomonas Genome database » se concentre sur les annotations de gènes et de n'offre pas d'amples informations relatives aux autres concepts biologiques où les protéines interviennent comme les voies métaboliques et les réactions enzymatiques. Cela pourrait être clairement remarqué si on compare, par exemple, l'entrée du gène «coxB » dans la base de données « Pseudomonas Genome database » (Locus* Tag: PA0105) et son entrée équivalente dans la base de données de Pseudomonas aeruginosa PAO1 de ***PseudomonsDW*** (ID: PAE02505). La première base de données ne donne aucune information sur les enzymes associées à la protéine codée par coxB. En outre, des informations sur les voies métaboliques où le produit du gène est impliqué sont limitées aux noms de ces voies et quelques liens vers la base de données KEGG. L'entrée de ***PseudomonasDW*** liste des sections spécifiques pour les enzymes et les voies métaboliques. Dans le cas de l'entrée de coxB dans ***PseudomonasDW***, elle fournit des informations riches sur l'enzyme sous-jacent relative à la protéine nommée cytochrome-c oxydase et deux voies auxquelles participe la protéine: la voie de la phosphorylation oxydative et la voie métaboliques.

D'autre part, ***PseudomonasDW*** fournit des informations sur un ensemble plus vaste d'espèces de Pseudomonas. Actuellement, 33 espèces sont intégrés où 10 d'entre eux ne s'affichent pas dans la base de données « Pseudomonas Genome database ». Ces espèces sont: pseudomonas aeruginosa M18, Pseuomonas aeruginosa NCGM2.S1, Pseuomonas aeruginosa 152504, Pseuomonas aeruginosa 138244, Pseudomonas putida BIRD-1, Pseudomonas putida S16, Pseuomonas stutzeri ATCC 17588, Pseuomonas stutzeri DSM 4166 et Pseudomonas chlororaphis.

Le fait d'étendre ***PseudomonasDW*** par un wiki biologique (PDWiki), ce qui n'est pas le cas dans la base de données « Pseudomonas Genome database », est de donner à la plate-forme la possibilité de migrer à partir d'un ensemble de bases de données biologiques classiques vers un très riche référentiel de connaissances pour les Pseudomonas où les données biologiques sont enrichies par la communauté d'annotations. Ceci permet aux utilisateurs de ***PseudomonasDW*** de collaborer entre eux en éditant et en ajoutant davantage des données pour la plate-forme et d'intégrer les connaissances spécialisées de nombreux groupes de chercheurs appartenant à des disciplines biologiques différentes.

# CONCLUSIONS ET PERSPECTIVES

# Conclusions et perspectives

Le genre Pseudomonas de la famille des Pseudomonaceae répond à la définition suivante : bacilles à Gram négatif, aérobies stricts à l'exception de certaines pouvant utiliser le NO3 comme accepteur d'électrons. Les Pseudomonas sont des bactéries ubiquitaires que l'on rencontre dans les sols, sur les végétaux et surtout dans les eaux douces et marines. Leur mobilité est assurée par plusieurs flagelles polaires, et elles ont un métabolisme mésophile et chimio-organothorphe, la plupart étant saprophytes. Quelques espèces comme P. syringae, sont phytopathogènes et certaines peuvent causer des infections chez l'humain. Particulièrement P. aeruginosa, reconnu comme pathogène opportuniste et causant des infections pulmonaires mortelles chez les patients atteints de fibrose kystique.

Vu l'importance biologique fournie par les Pseudomonas dans le domaine de la recherche, des études moléculaires approfondis ont été réalisées par les techniques d'études génomiques, dites à haut débit, qui génèrent un grand nombre d'informations. L'accumulation de ces informations dans des bases de données différentes a conduit à une hétérogénéité syntaxique et sémantique importante. De larges volumes de données sont actuellement disponibles publiquement, les types de données sont divers, et les ressources sont très nombreuse. Souvent les données provenant de différentes ressources présentent une hétérogénéité sémantique et syntaxique très importante.

L'hétérogénéité syntaxique se manifeste tout d'abord au niveau des formats pour décrire le contenu de sources. On trouve souvent le format ASN.1 (notation formelle pour décrire les données transmises lors de protocoles d'échanges), (e.g. Entrez), mais aussi des formats plus standard tels que XML (e.g. GenBank). A noter que les banques proposent souvent différents formats d'exportation de leurs données. Cette hétérogénéité de formats est accompagnée par une diversité des modèles de données : relationnel (e.g. Swiss-Prot), objet (e.g. Gus) ou semi–structuré (e.g. GenBank).

L'hétérogénéité sémantique recouvre plusieurs aspects. Elle concerne en premier lieu le focus. Chaque base se focalise sur un type d'objet biologique (e.g. le focus de swiss-Prot est la protéine, celui de GenBank est le gène, celui de PDB la structure 3D de la protéine). Aussi l'hétérogénéité sémantique est relative à la diversité des modes de désignation des entités. Différents vocabulaire sont utilisés pour annoter les séquences et la

confiance accordée à ces annotations est rarement totale. Par ailleurs, on retrouve pour une même entité (protéine ou gène) plusieurs noms, et ce, à l'intérieur d'une même banque.

Une autre forme de l'hétérogénéité provient des langages de requêtes. Souvent les langages sont de simples formulaires (combinaisons de mots à chercher dans un texte), dans le cas de portails ou de simples banques de données. Mais on peut aussi trouver des langages structurés tels que SQL (Genopage) ou OQL (Gus).

La grande diversité de ces données stockées, l'hétérogénéité des représentations, l'autonomie des sources les unes par rapport des autres, rendre difficile, voire impossible leur utilisation combinée par les biologistes. Aujourd'hui, l'un des grands défis de la bioinformatique est de permettre aux biologistes d'accéder efficacement à plusieurs sources de données ayant chacune un schéma global unifié via des procédures automatiques. Cette automatisation devrait aboutir à une véritable coopération entre le biologiste et la machine, pour une recherche plus efficace des informations et une meilleure exploitation des résultats.

Trois grandes approches pour l'intégration de sources d'information ont alors été proposées : les approches navigationnel, entrepôt et médiateur.

Dans l'approche entrepôt de données (approche matérialisée), les données sont extraites des différentes sources et combinées dans un schéma global. Par contre dans les deux autres approches (approche non matérialisée), les données restent au niveau des sources : ce sont des portails et des médiateurs.

L'intégration navigationnelle consiste à regrouper les bases de données entre elles à partir des identifiants qu'elles partagent. Il s'agit de la méthode la plus simple, accessible à tous les utilisateurs sans apprentissage préalable. Elle reprend le principe appliqué lors de l'extraction manuelle, en sélectionnant les attributs à extraire de chacune des sources demandées.

Les deux dernières approches, la construction d'un entrepôt de données ou l'intégration de données virtuelle à l'aide de vues ont besoin toutes les deux d'un modèle de données commun afin de représenter les données extraites des sources locales.

La démarche de création d'un entrepôt de donnés consiste à traduire massivement les données extraites des sources locales, afin de les rendre compatibles avec le modèle de données proposé à l'utilisateur. Cette adaptation des données présente un certain nombre d'inconvénients, tels que l'espace nécessaire au stockage et la mise à jour qui est très couteuse en temps et en trafic sur le réseau. Le système offre généralement un langage de requêtes qui permet d'appliquer des opérateurs d'extraction de données pour

La médiation de données permet d'intégrer uniquement les données souhaitées par l'utilisateur, qui exprime ses besoins au travers d'une requête posée sur un schéma global

préalablement défini. Les données sont à jour en permanence, puisque relues à chaque fois qu'une nouvelle demande parvient au système. L'espace demandé pour stocker les données est faible, et dédié au mécanisme de mise en cache des requêtes s'il a été mis en place par les concepteurs. Les difficultés majeures de la médiation reposent essentiellement sur la transformation de requêtes destinées aux sources de données locales, et la facilité d'évolution du schéma global en cas d'ajout ou de retrait d'une source, ce qui se produit très fréquemment sur le Web.

Dans ce cadre, notre travail a pour finalité la réalisation d'un environnement intégratif de données biologiques concernant les Pseudomonas. Ce travail entre dans le cadre d'une collaboration entre notre laboratoire de recherche LABIPHABE et le groupe KHAOS de l'université de Malage.

Dans cette thèse, nous nous sommes intéressés au problème d'intégration de données sur le Web, en nous focalisant particulièrement sur les problèmes posés par les sources de données biologiques. Les deux derniers chapitres de ce mémoire s'articulent autour de la mise en œuvre d'un système intégratif pour l'intégration de données biologiques :

Les deux premiers chapitres mettent en évidence les différentes caractéristiques des sources de données biologiques et comportent une description des divers niveaux d'hétérogénéité entre les sources. Ils dressent aussi un état de l'art qui illustre chacune des solutions majoritairement suivies en informatique (entrepôt, médiateur et système navigationnel) et montrent comment elles ont été appliquées aux données biologiques.

Dans le troisième chapitre nous avons proposé une approche hybride, qui combine entre les avantages de l'architecture entrepôt de données et celle de médiateur, pour une intégration de données forte et efficace. Cette approche a été adaptée au domaine biologique afin de proposer une solution d'intégration simple et flexible.

Le quatrième chapitre a été conçu pour décrire une plateforme complète qui offre des informations allant du gène à la voie métabolique et qui réconcilie ces données afin d'avoir une vue unifiée des informations disponibles sur une protéine donnée.

# 1 RÉSUMÉ DES CONTRIBUTIONS

Conscients du fait que les sources biologiques aujourd'hui ouvertes sur le Web ne fournissent pas encore les métadonnées, ou ne garantissent pas les droits nécessaires à leur exploitation de façon aisée par le biais de procédures (semi-automatisées), nos travaux se sont concentrés sur la résolution d'une classe de problèmes d'intégration qui se rencontrent

principalement à l'échelle individuelle : l'objectif visé étant d'automatiser autant que possible les phases d'interrogation des sources de données biologiques hétérogènes, divers et reparties sur le web et de réconciliation des résultats partiels. Les contributions de nos travaux concernent plusieurs points :

**Adaptation d'une approche hybride pour l'intégration sématique des données biologiques de Pseudomonas Sp**

La quantité des données issues de l'étude biotechnologique de l'espèce de Pseudomonas requérant un accès à une grande diversité de données réparties dans de multiples sources de données. Nous avons donc opté pour le développement d'un entrepôt de données et ainsi proposé des solutions pour une intégration systématique et réconciliée de données hétérogènes.

PseudmonasDW est un entrepôt de données semi-structuré pour stocker, gérer, et intégrer les informations biologiques collectées de sources de données via le Web. PseudmonasDW se focalise sur l'intégration de données de pseudomonas sp.

Pour la conception du système PseudmonasDW, nous avons utilisé le processus d'intégration qualifié d'ascendant (ou bottom-up) où nous sommes partis du besoin de représenter au sein d'un même schéma les données souhaitées, pour ensuite choisir les sources de données ainsi que le processus d'intégration appropriés. PseudomonasDW intègre des données génomiques, protéiques, enzymatiques et métaboliques à partir de cinq sources de données divers et réparties sur le web : Genbank, PRODORIC, Uniprot, BRENDA et KEGG.

Ainsi, pour l'intégration les données, nous avons combiné les deux approches matérialisé et virtuelle pour exploiter leurs avantages dans un nouveau environnement hybride. Dont nous avons utilisé les services de données pour extraire et transformer les données collectées à partir des sources de données. Les adaptateurs forment une partie importante dans les services de données qui fournissent des moyens pour interroger et corréler les différents types d'informations intégrés. Les services de données initialisent le processus d'ETL, dont les adaptateurs sont considérés comme une interface qui reçue des requêtes XQuery, interroge les sources de données, extraite les données souhaités et les transforme en un modèle commun utilisé par le SB-KOM. La sémantique de nos services de données inclut des informations sur le schéma de la source et la provenance de données. Contrairement à l'entrepôt de données GEDAW, cité dans la partie introductive de ce manuscrit, garder la traçabilité et la provenance de données est nécessaire, dans le domaine de la bioinformatique, dont il est très important de savoir quelle source de données a été utilisée dans l'extraction d'une telle donnée. Nous avons développé cinq services de données : un service pour une source de données.

PseudomonasDW intègre des sources de données offrant des informations chevauchantes. Une agrégation d'information a été alors requise pour identifier des objets

équivalents d'un point de vue sémantique. Nous avons appliqué une intégration sémantique pour supprimer toute redondance au niveau du schéma de l'entrepôt. L'intégration sémantique dans PseudomonasDW est fondée sur la construction d'un schéma global intégrateur et vise à convertir les données des sources en termes des données dans ce schéma global intégrateur.

Dans PseudmonasDW, nous avons suivi l'approche GAV (Global-As View) qui consiste à définir le schéma global en fonction des schémas locaux des sources de données. Notre propose était l'utilisation d'une ontologie (PseudomonasDW Ontology) comme un schéma global de l'entrepôt. Notre ontologie a été construite par la réconciliation de tous les différents schémas de sources en une seule ontologie cohérente.

L'ajout d'une source de données exige une modification profonde du schéma global de PseudomonasDW. Contrairement aux entrepôts de données GenMapper et GeWare, cités dans la partie introductive de ce manuscrit, qui sont adaptés à l'ajout de nouvelle sources de données par l'utilisation du modèle générique GAM. Ce modèle modélise les sources de données plutôt que leur contenu. La modification de schéma global au niveau de GenMapper et GeWareest considérée comme une extension du schéma plutôt qu'une modification profonde.

Les différents composants du SB-KOM (contrôleur, planificateur de requête et l'évaluateur/intégrateur) participent dans le processus ETL dans PSeudomonasDW. Le médiateur est basé sur le répertoire sémantique SD-Core dans lequel nous avons enregistré notre ontologie, les schémas des sources et nos règles de correspondances. Le SD-Core a joué le rôle du middleware entre PseudomonasDW et le SB-KOM.

Les instances de notre schéma intégrateur servent d'étape de transformation préalable au peuplement de PseudomonasDW. L'utilisation de l'ontologie et des instances permet l'inclusion de raisonnement aux différents niveaux. Les différentes instances retournées par le SB-KOM sont chargées dans PseudmonasDW après une translation automatique en XML par le biais de quelques bibliothèques du Java. L'utilisation d'un système médiateur pour une intégration sémantique de données dans un entrepôt de données nous a permis d'exploiter leurs avantages dans une nouvelle approche. D'une part, les données sont physiquement stockées dans l'entrepôt pour être prêtes à une interrogation directe et rapide. Et d'autre part, l'intégration et la mise à jour des données sont virtuellement achevées en utilisant le médiateur.

Les bases de données UniProt et GenBank créent des listes de diffusion. Ces listes sont destinées à la distribution des messages qui annoncent les mises à jour effectuées au niveau de ces deux bases de données. L'abonnement à ces listes nous a permis de recevoir les dernières modifications et de garder une trace des mises à jour des entrées individuelles.

Les sources de données PRODORIC, BRENDA et KEGG sont périodiquement mis à jour et fournissent des archives complètes qui contiennent uniquement les entrées

actualisées. Ces archives nous ont permis de spécifier quelles entrées intégrées dans PseudomonasDW ont été mis à jour. Lorsque le système est informé par les entrées modifiées, la mise à jour des données est pratiquement intégrée à l'aide du SB-KOM.

Nous avons développé un module Java qui génère des requêtes conjonctives et les envoie au système SB-KOM pour performer les processus d'extraction et de transformation. SB-KOM fait appel aux services de Web que nous avons développé pour extraire uniquement les données modifiées à partir des entrées originales. Par la suite, il est possible de lancer automatiquement le processus d'intégration pour mettre à jour l'entrepôt de données en remplaçant seulement les données obsolètes par elles actualisées.

Dans PseudomonasDW, le système est une plate-indépendant et n'exige aucune installation local. Il est disponible pour l'utilisateur via une interface Web contrairement à certains entrepôts exemple de BioWarehouse qui est un système linux-dépendant et exige une installation locale. Cela rendre l'utilisation de ce type de système une tâche fastidieuse surtout pour les biologistes qui ne maîtrisent pas l'outil informatique et particulièrement la plateforme Linux.

Avec PseudmonasDW, nous aimerions fournir aux biologistes un outil accessible pour élucider les processus cellulaire d'intérêt en utilisant une stratégie de système intégré.

**Développement d'une plateforme Biologique pour les Pseudomonas**

Pour le développement des bases de données de PseudomonasDW, nous nous sommes basés sur les approches qui abordent la problématique de l'entreposage de documents XML. Nous avons perçu un entrepôt XML comme une collection de documents XML qui contiennent les données extraites. Nous avons utilisé eXist pour stocker nos documents XML dans des bases de données natives. eXist, nous a permis de charger automatiquement (en utilisant les différentes ses différentes options) les documents XML dans 33 collections : une collection pour chaque espèce entreposé dans PseudomonasDW.

Dans le but de faciliter et d'accélérer le processus d'interrogation des bases de données de PseudomonasDW, nous avons développé des indexes qui sont créés et maintenus automatiquement dans eXist. Nous avons suivis la nouvelle procédure d'indexation basée sur les noms des éléments. Cela nous a permis de retrouver facilement tous les éléments d'un certain nom quelle que soit leur imbrication.

Les bases de données de PseudomonasDW sont publiquement accessibles via une interface Web disponible sur le lien http://www.pseudomonasdw.khaos.uma.es . C'est une application web que nous avons développé en utilisant principalement quelques technologies du Web et de Java (JSP, Java, Servlet API, XHTML, CSS, XSLT, JavaScript, JQuery). L'application Web est implémentée sur le serveur Web Apache 2.0.

L'interface utilisateur de PseudomonasDW incorpore des outils bioinformatiques pour permettre aux utilisateurs d'analyser et comparer les données stockées. Nous avons incorporé l'outil GBrowse qui permet la navigation dans les génomes et leur visualisation, il affiche une représentation graphique d'une section d'un génome, ainsi que les positions des gènes en plus d'autres éléments fonctionnels. Nous avons intégré aussi l'outil Blast qui est un programme permettant de réaliser des alignements et des comparaisons locaux entre deux séquences (nucléiques ou protéiques).

PseudomonasDW contient 170000 entrés et fournit des informations sur un ensemble très vaste d'espèces de Pseudomonas. Actuellement, 33 espèces sont intégrés où 10 d'entre eux ne s'affichent pas dans la base de données « Pseudomonas Genome database ». Ces espèces sont: pseudomonas aeruginosa M18, Pseuomonas aeruginosa NCGM2.S1, Pseuomonas aeruginosa 152504, Pseuomonas aeruginosa 138244, Pseudomonas putida BIRD-1, Pseudomonas putida S16, Pseuomonas stutzeri ATCC 17588, Pseuomonas stutzeri DSM 4166 et Pseudomonas chlororaphis.

La base de données « Pseudomonas Genome database » ne donne aucune information sur les enzymes associées à la protéine. En outre, des informations sur les voies métaboliques où le produit du gène est impliqué sont limitées aux noms de ces voies et quelques liens vers la base de données KEGG. L'entrée de PseudomonasDW liste des sections spécifiques pour les enzymes et les voies métaboliques.

Le fait d'étendre ***PseudomonasDW*** par un wiki biologique (PDWiki), ce qui n'est pas le cas dans la base de données « Pseudomonas Genome database », est de donner à la plate-forme la possibilité de migrer à partir d'un ensemble de bases de données biologiques classiques vers un très riche référentiel de connaissances pour les Pseudomonas où les données biologiques sont enrichies par la communauté d'annotations. Ceci permet aux utilisateurs de ***PseudomonasDW*** de collaborer entre eux en éditant et en ajoutant davantage des données pour la plate-forme et d'intégrer les connaissances spécialisées de nombreux groupes de chercheurs appartenant à des disciplines biologiques différentes.

# 2 OUVERTURE ET PISTES DE RECHERCHE

La récente expansion des sources de données biologiques sur le Web les a mises à disposition d'un nombre sans cesse croissant de chercheurs, ouvrant ainsi de très nombreuses perspectives d'innovation. La biologie a ainsi pris une nouvelle dimension : anciennement divisée en plusieurs disciplines, elle est devenue intégrative et offre désormais de belles perspectives d'appréhension de la complexité du monde vivant. L'intégration de données vise à combler le fossé qui existe entre producteurs et consommateurs de données, particulièrement dans ce domaine. Dans le cadre de cette thèse, nous avons orienté nos recherches afin de rapprocher ces différents acteurs.

Nous pensons améliorer à court terme les travaux que nous avons exposés, en nous focalisant sur plusieurs points particuliers :

- Concernant l'architecture de l'entrepôt ***PseudomonasDW*** :
  - ✓ Associer des méta-données décrivant plus précisément la confiance accordée à la source et sa qualité estimée.
  - ✓ Développement d'un algorithme de mise à jour pour garantir la performance des données stockées au niveau de ***PseudomonasDW***.
  - ✓ Automatiser la recherche de correspondance entre éléments des schémas locaux des sources et le schéma global de l'entrepôt pour rendre l'ajout des nouvelles sources de données plus facile.
- Concernant l'intégration des données :
  - ✓ Intégrer non seulement des sources de données, mais aussi des services Web : cette technologie s'est grandement développée ces dernières années dans le domaine biologique, et les perspectives offertes semblent très prometteuses
  - ✓ Associer notre entrepôt de données à des méthodes d'analyse et de prédiction plus évoluées que celles que nous avons utilisées pour fouiller et comparer les données intégrées.

# GLOSSAIRE

# Glossaire

**Acide aminé :** Monomère constitutif des protéines. Il en existe 20, codés par un système à trois nucléotides (codons), dans l'ARN.

**ADN (Acide DésoxyriboNucléique) :** L'ADN est la forme de stockage de l'information génétique du génome de tous les êtres vivants. Cette information est représentée sur le chromosome par une suite linéaire de gènes, séparés par des régions intergéniques. L'ADN, macromolécule biologique formée de désoxyribonucléotides, est un des constituants des chromosomes. Les molécules d'ADN s'étirent en un très long fil, constitué par un enchaînement (séquence) précis d'unités élémentaires que sont les nucléotides. La structure originale de l'ADN, formée de deux brins complémentaires enroulés en hélice (double hélice), lui permet de se dupliquer en deux molécules identiques entre elles et identiques à la molécule mère lors du phénomène de réplication.

**Agrégation :** Action de calculer les valeurs associées aux positions parents des dimensions hiérarchiques. Cette agrégation peut être une somme, une moyenne, ou tout autre processus plus complexe.

**Annotation :** L'annotation du génome consiste à prédire et localiser l'ensemble des séquences codantes (gènes) du génome c'est-à-dire à déterminer et identifier leur structure (annotation syntaxique ou structurale), leur fonction (annotation fonctionnelle) ainsi que les relations entre les entités biologiques relatives au génome (annotation relationnelle). L'information résultante enrichit les sources de données biologiques.

**API (Application Programming Interface) :** Interface pour langages de programmation, matérialisées par des primitives, permettant à une application d'accéder à des programmes système pour, par exemple, communiquer ou extraire des données.

**ARN (Acide RiboNucléique) :** L'ARN est une macromolécule biologique formée de ribonucléotides permettant de stocker et de traiter l'information dans la cellule. L'ARN est une séquence d'acide nucléique linéaire, simple brin. On distingue les ARN messagers, ARN de transfert, les ARN ribosomaux, les ARN nucléaires et les ARN cytoplasmiques.

**Blast :** Initialement Blast est un outil de recherche d'informations dans les banques de séquences, comportant un algorithme de comparaison de séquences. Aujourd'hui, on utilise le terme Blast pour dénoter uniquement l'algorithme de comparaison de séquences. Il existe de nombreuses versions d'algorithmes Blast de comparaisons de séquences à travers les sources. Il existe des Blasts qui permettent la comparaison de séquences d'acides aminés donc de comparer les séquences des protéines et d'autres qui comparent les séquences de nucléotides dont sont constitués les gènes. Certaines des versions disponibles sont dotées d'heuristiques, de paramètres, et d'autres non.

**Chromosome :** Ensemble d'éléments d'information liés entre eux dans une même molécule d'ADN. (en biologie cellulaire) : le chromosome est une structure cytologique résultant d'une hypercondensation de la chromatine, permettant la réparation du matériel génétique entre les cellules filles lors de la mitose ou de la méiose. Chromosome vient de "chromos", couleur : allusion

à leur capacité de fixer les colorants. Les chromosomes ne sont visibles, en général, que durant la division cellulaire.

**Cluster :** (grappe en français) Architecture de groupes d'ordinateurs, utilisée pour former de gros serveurs. Chaque machine est un nœud du cluster, l'ensemble est considéré comme une seule et unique machine, permettant d'obtenir une grande puissance de traitement. Ce type d'architecture est utilisé principalement pour le décisionnel, le transactionnel et l'entrepôt de données.

**Data Mart** : Base de données orientée sujet mise à disposition des utilisateurs dans un contexte décisionnel décentralisé.

**Dimension :** Axe d'analyse correspondant le plus souvent aux sujets d'intérêt de l'entrepôt de données; exemple: dimension temporelle, dimension protéique …

**Drill-down :** Consiste à représenter les données du cube à un niveau de granularité supérieur conformément à la hiérarchie définie sur la dimension. Une fonction d'agrégation (somme, moyenne, …) spécifiée pour la mesure et la dimension indique comment sont calculées les valeurs du Niveau supérieur à partir de celles du niveau inférieur.

**DTD :** Une DTD, acronyme anglais signifiant Document Type Definition, se traduisant par Définition de Type de Document, est un document permettant de décrire un modèle de document SGML ou XML. Une DTD indique les noms des éléments pouvant apparaître et leur contenu constitué par leurs sous-éléments et leurs attributs.

**Espèce :** Ensemble d'individus ayant des caractéristiques génétiques semblables. Chez les organismes à reproduction sexuée, les individus sont interféconds : le produit de leur croisement est fertile. Chez les procaryotes, l'unité repose sur les similitudes du génome et du phénotype.

**Eucaryote :** Organisme vivant dont les cellules possèdent un noyau au sein duquel est isolé le génome nucléaire.

**Expression génique (Gene expression) :** L'expression génique (énoncée dans le dogme central de la biologie moléculaire) englobe les différentes étapes conduisant du gène aux protéines, notamment celles de transcription et de traduction. Elle est sous le contrôle de divers mécanismes de régulation.

**Fait :** Objet d'analyse dans le cadre d'un modèle multidimensionnel, souvent une donnée numérique.

**FASTA :** Un outil d'alignement de séquences ADN ou protéiques proposé par David J. Lipman et William R. Pearson en 1985 dans l'article "Rapid and sensitive protein similarity searches". Le programme original "FASTP" était destiné à la recherche de similarités entre protéines.

**Gène :** Le gène est un segment d'ADN situé à un endroit bien précis (locus) sur un chromosome et porteur d'une information génétique.

**Génome :** Ensemble du matériel génétique (patrimoine héréditaire) d'un individu ou d'une espèce. Il est constitué de molécules d'acides nucléiques (ADN ou ARN). Les gènes, c'est-à-dire les parties d'ADN porteuses d'une information génétique, ne constituent qu'une partie du génome.

**GNU (GNU's Not UNIX) :** Projet de la Free Software Foundation visant à concevoir, réaliser et distribuer un système d'exploitation libre et complet inspiré d'Unix.

**HTML (HyperText Markup Language) :** Langage de description de pages Web. Un standard initié par le W3C et compatible tous systèmes.

**Internet :** INTERconnected NETworks. Réseau international de réseaux interconnectés.

**Interopérabilité :** c'est le fait que plusieurs systèmes, qu'ils soient identiques ou radicalement différents, puissent communiquer sans ambiguïté et opérer ensemble.

**Intron :** Partie du gène située entre deux exons et dont le rôle est encore inconnu. L'ARN correspondant aux introns est excisé par épissage de l'ARN précurseur lors de sa maturation.

**Locus :** Localisation (site) précise sur le chromosome (peut être un gène ou toute autre position choisie).

**Modèle de données :** Ensemble de règles permettant de formaliser le monde réel sous la forme d'un schéma de données.

**MOLAP (Multidimensionnal On Line Analytical Processing) :** Equivalent à OLAP, utilisant une base de données multidimensionnelle. Pour le premier, les jointures sont déja faites, ce qui explique les performances. Dans le second, les jointures entre les tables de dimension et de fait sont effectuées au moment de la requête.

**OLAP (On Line Analytical Processing):** Caractérise l'architecture nécessaire à la mise en place d'un système d'information décisionnel. S'oppose à OLTP. Le terme OLAP désigne souvent une catégorie d'outils d'exploration de données qui permettent de visualiser des valeurs dans plusieurs dimensions.

**Oligonucléotide :** Petit segment d'ADN (quelques dizaines de nucléotides) simple brin.

**OLTP (On Line Transactionnel Processing):** Type d'environnement de traitement de l'information dans lequel une réponse doit être donnée dans un temps acceptable et consistant.

**Opéron :** Unité de transcription constituée par un promoteur (courte séquence nécessaire à l'initiation de la transcription), un opérateur (site auquel un répresseur se lie, pour empêcher le déclenchement de la transcription) et un ou plusieurs gènes.

**OQL (Object Query Language) :** Langage d'interrogation de bases de données objet proposé par l'ODMG ; il est fondé sur une extension de SQL supportant chemins, méthodes, héritage et collections.

**Perl :** un langage optimisé pour extraire des informations de fichiers texte et imprimer des rapports basés sur ces informations. C'est aussi un bon langage pour de nombreuses tâches d'administration système. Il est écrit dans le but d'être pratique (simple à utiliser, efficace, complet) plutôt que beau (petit, élégant, minimaliste). Perl combine les meilleures fonctionnalités de C, sed, awk et sh, de manière telle que les personnes familières de ces langages ne devraient avoir aucune difficulté avec celui-ci.

**Phénotype :** L'expression visible de l'action des gènes. Il englobe tout ce qui est anatomique (physique extérieur, visible de tous, comme le physique intérieur de chaque être) et physiologique notamment. Un comportement particulier, tout comme une combinaison de comportements, peuvent également être considérés comme des phénotypes, résultant de l'association d'un ou plusieurs gènes. En réalité, le phénotype n'est pas seulement du au génotype (c'est-à-dire aux gènes et à leur expression). Il est également du à l'action du milieu dans lequel vit l'individu. En fait, un caractère peut être génétiquement déterminé, mais il se peut qu'il ne s'exprime en réalité pas ou moins selon le milieu. (Prenons un exemple, hors comportement animal : le diabète génétiquement déterminé. L'individu développera la maladie ou non selon le milieu et en cas selon son alimentation. En cet exemple-ci, l'influence du milieu prime sur celle du génotype. Mais l'inverse existe également.).

**Plug-in :** Aussi appelé « greffon ». Logiciel tiers venant se greffer à un logiciel principal afin de lui apporter de nouvelles fonctions. Le logiciel principal fixe un standard d'échange d'informations auquel ses greffons se conforment. Le greffon n'est généralement pas conçu pour fonctionner seul.

**Protéine :** La protéine est un produit du gène issu de la synthèse protéique via le code génétique. Les protéines sont des macromolécules constituées de longues chaînes d'acides aminés (de 50 à 30000 acides aminés, la moyenne étant d'environ 400) qui se replient sur elles-même et adoptent des conformations très spécifiques dans l'espace. L'ensemble des protéines codées sur le génome (= le protéome) peut être ainsi considéré comme une collection de repliements 3D suffisants pour assurer les principales fonctions cellulaires, comme le métabolisme, la réplication ou la gestion de l'information.

**Puce à ADN :** Technique d'hybridation permettant une analyse génomique comparative (i.e. une comparaison globale) de l'expression d'un grand nombre de patterns d'ARNm. Immobilisés sur un support solide (matrice), des oligonucléotides (simples brins) spécifiques de différents gènes ou ADNc connus constituent les sondes dont le rôle est de détecter des cibles marquées complémentaires, présentes dans le mélange complexe à analyser (ARNm extraits de cellules, tissus ou organismes entiers et convertis en ADNc). Les sondes sont soit greffées sur le support, soit synthétisées in situ (unité d'hybridation = plot). Les signaux d'hybridation sont détectés selon le type de marquage, radioactivité ou fluorescence, par mesure radiographique ou par fluorescence, et quantifiés.

**Puce à CGH :** La technique d'hybridation génomique comparative (CGH), permet de caractériser les gains et pertes de segments chromosomiques qui ont lieu dans les cellules cancéreuses. Le principe d'une puce à CGH est, comme la puce à ADN, fondé sur l'hybridation. Dans une puce à CGH, on dépose sur une matrice une représentation complète d'un génome sain, chaque spot contenant un BAC marqué par un fluorochrome rouge. On hybride alors la puce avec un ADN tumoral, marqué par un fluorochrome vert. Si dans la tumeur un segment chromosomique était sur-représenté, il y aura un excès d'ADN vert correspondant à ce segment, et après hybridation du mélange de sondes, le segment chromosomique correspondant sera plus vert que rouge. De manière symétrique, si un segment chromosomique était perdu dans la tumeur, le segment correspondant du chromosome normal sera plus rouge que vert. Cette technique permet ainsi de caractériser, avec une résolution d'environ 10-20 mégabases, l'ensemble des gains et pertes présents dans une tumeur donnée, et où pourraient se trouver localisés respectivement des oncogènes et des suppresseurs de tumeurs.

**Puce à protéines :** Système permettant l'analyse de l'ensemble des protéines synthétisées à partir du génome. Des quantités de protéines de l'ordre de la femtomole ($10^{-15}$ M) sont déposées sur un support métallique et analysées par spectrométrie de masse.

**ROLAP (Relational On Line Analytical Processing):** Cette technique permet de faire de l'analyse multidimensionnelle à partir de données stockées dans des bases relationnelles.

**Roll-up :** Consiste à représenter les données du cube à un niveau de granularité inférieur, donc sous une forme plus détaillée.

**Sémantique :** La sémantique est, dans les sciences du langage, opposée à la syntaxe. La syntaxe concerne les règles formelles, alors que la sémantique concerne la signification. Dans le domaine informatique, le but du "Semantic Web" est de permettre aux machines d'échanger des informations en utilisant le sens des mots comme dans les langages naturels. Cet objectif ambitieux nécessite un travail important sur les langages, la structure des systèmes, et les ontologies.

**Séquençage :** Détermination de l'ordre linéaire des composants d'une macromolécule (les acides aminés d'une protéine, les nucléotides d'un acide nucléique, etc.). Le séquençage de l'ADN ("décryptage" du génome) s'effectue selon le protocole enzymatique de Sanger. Séquençage d'étiquettes (signature sequencing) : pour identifier un gène, on n'utilise que la séquence d'un petit fragment, ou étiquette (tag), correspondant à la signature des gènes.

**Séquence :** Succession de monomères dans un polymère. L'orientation de la séquence est définie par la synthèse du polymère. Les séquences nucléiques (ADN ou ARN) sont des polynucléotides (polymères de nucléotides).

**Service Web :** Technologie permettant à des applications de dialoguer à distance via Internet indépendamment des plates-formes et des langages sur lesquelles elles reposent.

**SGBD (Système de Gestion de Bases de Données) :** Un SGBD est une collection de logiciels permettant de créer, de gérer et d'interroger efficacement une base de données indépendamment du domaine d'application.

**Spectrométrie de masse :** Une technique d'analyse chimique permettant de détecter et d'identifier des molécules d'intérêt par mesure de leur masse monoisotopique. De plus, la spectrométrie de masse permet de caractériser la structure chimique des molécules en les fragmentant. Son principe réside dans la séparation en phase gazeuse de molécules chargées (ions) en fonction de leur rapport masse/charge (m/z). La spectrométrie de masse est utilisée pratiquement dans tous les domaines scientifiques : physique, astrophysique, chimie en phase gazeuse, chimie organique, dosages, biologie, médecine...

**SQL (Structured Query Language) :** Langage de requête de base de données et de programmation largement utilisé pour accéder à, interroger, mettre à jour et gérer des données dans des systèmes de bases de données relationnelles. En utilisant le langage SQL, l'utilisateur peut extraire des données d'une base de données, créer des bases de données et des objets de base de données, ajouter des données, modifier des données existantes et exécuter d'autres fonctions plus complexes. SQL donne également la possibilité de modifier la configuration d'un serveur, de

modifier des paramètres de base de données ou de session et de contrôler les instructions de données et d'accès.

**Taxonomie :** Science des lois de la classification des formes vivantes. Elle inclut la reconnaissance, l'identification des formes vivantes et leur rangement dans une classification.

**Transcriptome :** Ensemble des ARN messagers transcrits à partir du génome.

**URL :** Cet acronyme signifie Uniform Resource Locator, qui se traduit littéralement par localisateur uniforme de ressource, et désigne une chaîne de caractères (codée en ASCII, donc utilisant l'alphabet anglais, ce qui signifie qu'elle ne présente aucun accent comme é ou î) qui est utilisée pour adresser les ressources du World Wide Web telles que des documents HTML, des images ou des sons.

**Web :** Système basé sur des liens hypertextes, permettant l'accès aux ressources du réseau Internet.

**Web sémantique :** N'est pas un Web distinct mais bien un prolongement du Web que l'on connaît et dans lequel on attribue à l'information une signification clairement définie, ce qui permet aux ordinateurs et aux humains de travailler en plus étroite collaboration.

**XML (eXtensible Markup Language) :** Standard du W3C qui permet de décrire les données et de les structurer de telle sorte qu'elles puissent être échangées entre un large nombre d'applications en différents environnements hardware et software.

**Xquery (XML Query) :** Langage de requête permettant d'accéder à chacun des éléments d'information d'un document XML, d'en sélectionner des listes et de les manipuler. XQuery est un sur-ensemble de XPath.

# ANNEXES

# Annexe 1 : UML

La notation UML est un langage de modélisation dont la première version date de 1996. UML est une norme de l'OMG (Object Management Group) qui est un consortium des principaux constructeurs et éditeurs de logiciels. La notation UML se veut intuitive, homogène, cohérente (élimination des symboles embrouillées ou redondants) et d'une sémantique précise : tout cela doit faciliter les échanges entre les différents intervenants. UML ne cherche pas la spécification à outrance : en cas de besoin, des précisons peuvent être apportées par des mécanismes d'extension et/ou des commentaires en texte libre. UML défini 6 modèles pour la représentation des points de vues de la modélisation des systèmes informatiques :

- **Modèle des cas d'utilisation** : décrit les besoins de l'utilisateur,
- **Modèle des classes** : capture la structure statique,
- **Modèle d'interaction** : représente les scénarios et les flots de messages,
- **Modèle des états** : exprime le comportement dynamique des objets,
- **Modèle de déploiement** : précise la répartition des processus,
- **Modèle de réalisation** : montre les unités de travail

Ces modèles sont manipulées grâce à des diagrammes, ceux-ci pouvant correspondre à des vues complètes ou partielles des diagrammes. Il existe 14 sortes de diagrammes.

- **Diagramme des classes** : structure statique, il représente les classes intervenant dans le système,
- **Diagramme des états/transitions** : comportement d'une classe en termes d'états,
- **Diagramme d'objets** : représentation des objets (des occurrences des classes) et de leur relations, ils correspondent à des diagrammes de collaboration simplifiés (sans envoi de message),

- **Diagramme des paquetages**: un paquetage étant un conteneur logique permettant de regrouper et d'organiser les éléments dans le modèle UML, le Diagramme de paquetage sert à représenter les dépendances entre paquetages, c'est-à-dire les dépendances entre ensembles de définitions,
- **Diagramme de structure composite**: permet de décrire sous forme de boîte blanche les relations entre composants d'une classe,
- **Diagramme de séquences** : représentation temporelle des objets et de leurs interactions,
- **Diagramme de communication** : représentation simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets,
- **Diagramme global d'interaction**: permet de décrire les enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences,
- **Diagramme de temps**: permet de décrire les variations d'une donnée au cours du temps,
- **Diagramme des cas d'utilisation** : il permet d'identifier les possibilités d'interaction entre le système et les acteurs, c'est-à-dire toutes les fonctionnalités que doit fournir le système,
- **Diagramme d'activités** : représentation du comportement d'une opération en termes d'actions,
- **Diagramme de composants** : représentation des composants physiques d'une application,
- **Diagramme de profile** : utilise au niveau de méta-modèle où il représente les stéréotypes des classes ou des packages,
- **Diagramme de déploiement** : représentation du déploiement des composants sur les dispositifs matériels.

# Annexe 2 : Bases de données natives

Le terme *Native XML Database* (NXD), ou base de données XML native, est apparu pour la première fois dans une campagne de publicité, une base de données XML native de Software AG (Schöning, 2001). Grâce au succès de cette compagne, le terme est arrivé dans l'usage courant par différentes entreprises développant des produits similaires. Etant devenu un terme publicitaire, il n'a jamais eu de définition technique formelle. Une définition possible de ce qu'est une base de données XML native serait la suivante :

- Une base de données XML native définit un modèle logique pour un document XML. Elle stocke et récupère les documents suivant ce modèle de données. Au minimum, il doit inclure les éléments, les attributs, les données et l'ordre du document.
- Une base de données XML native gère le document XML comme une unité fondamentale de stockage, comme une ligne dans une table relationnelle.
- Les bases de données XML natives n'ont pas un modèle physique sous-jacent particulier. Par exemple, le modèle physique peut être relationnel, hiérarchique, orienté objet ou utiliser un format de stockage propriétaire comme des fichiers compressés indexés.

La première partie de cette définition est similaire à celle des autres types de bases de données, définissant le modèle utilisé pour le stockage et l'interrogation. Il existe un certain nombre de modèles pour XML comme DOM. Le modèle choisi pour faire une base de données XML native doit être conçue pour supporter arbitrairement la profondeur de l'imbrication des nœuds, la complexité de leurs relations, leur ordre, leur identité, etc.

La seconde partie de cette définition explique que l'unité de stockage fondamentale dans une base de données native XML est le document XML. Bien qu'il semble possible qu'une base de données XML native puisse assigner ce rôle à des fragments de documents, l'unité de stockage fondamentale reste effectivement le document XML dans la plupart des bases de données XML actuelles.

La troisième partie de la définition montre que le modèle physique sous-jacent n'est pas important. C'est exact et c'est certainement le cas pour toutes les sortes de base de

données. Le format de stockage physique utilisé par une base de données relationnelle n'est pas une condition nécessaire au caractère relationnel de la base. De plus, il est tout à fait envisageable d'utiliser un support relationnel pour fabriquer un moteur de base de données XML native comme eXist l'a fait à ses débuts.

Les bases de données XML natives sont donc des bases données conçues spécialement pour stocker des documents XML et comme les autres bases de données, elles gèrent les transactions, la sécurité, l'accès multi-utilisateurs, offrent des API de programmation, des langages de requêtes, etc. Les bases de données XML natives s'inscrivent donc parfaitement dans notre approche entièrement basée sur XML.

# Annexe 3 : eXist, une base de données XML native libre

Le projet eXist est une implémentation libre (LGPL) d'un système de gestion de base de données XML native, inerfaçable entre autres à l'aide de XPath, de XQuery et de XUpdate. Le projet a été entamé en 2000 par Wolfgang Meier, un développeur allemand. Il s'est basé sur les travaux de Shin, Jang et Jin (Shin, et al., 1998) qui proposaient un système efficace d'indexation des documents structurés. Ce fut tout d'abord une expérience d'implémentation d'une indexation de documents XML à l'aide d'un système relationnel. Aujourd'hui, eXist n'utilise plus de relationnel et fonctionne sur un système de stockage propre. La communauté autour d'eXist ne cessant de croître et les développeurs étant très actifs, eXist est devenu un SGDB XML natif complet. La base de données est complétement écrite en Java et peut être déployée de multiple façons, aussi bien comme un processus serveur que dans un moteur de *servlet* ou encore directement intégré dans une application.

eXist fournit un stockage sans schéma des documents XML dans des collections hiérarchiques. Une collection est un ensemble qui peut contenir d'autres collections ou des documents XML. En utilisant une syntaxe étendue d'XPath et d'XQuery, les utilisateurs peuvent interroger différentes parties de la hiérarchie de collections, ou tous les documents contenus dans la base de données. Le moteur de requêtes d'eXist implémente un traitement de requête efficace et basé sur les indexes. Le plan d'indexation permet une identification rapide des relations structurelles entre les nœuds, comme la relation parent-enfant, ancêtre-descendant et frère-suivant, frère-précédent. Basée sur des algorithmes de jointures de chemins, une large fourchette d'expressions de chemin est traitée en utilisant uniquement les informations d'index. L'accès aux nœuds courants, stockés dans le magasin central de documents XML, n'est pas nécessaire pour ce type d'expressions.

La base de données convient bien aux applications manipulant des petites ou larges collections de documents XML qui sont occasionnellement mises à jour. Le logiciel a été conçu de sorte qu'il supporte les documents orientés données ou présentation. Cependant, l'interrogation de ces derniers n'est pas très bien supportée par les langages de requêtes XML comme XPath. eXist fournit donc un certain nombre d'extensions au standard XPath

et XQuery pour traiter efficacement des requêtes de recherche textuelle, incluant entre autres la recherche par mot clé ou via des expressions régulières.

**Architecture d'eXist**

eXist est bel est bien un système de gestion de base de données XML natif, conformément à notre définition vue à la section 3.1. En effet, un modèle logique pour les documents XML est définit et le document XML est son unité de stockage fondamentale.

Les détails d'implémentation concernant le stockage des données sont totalement séparés du corps d'eXist (Figure 53). Tous les appels au système de stockage se font par des courtiers (Brokers). Un courtier peut être vu comme une interface entre le cœur d'eXist et les systèmes de stockages. Ces classes courtiers fournissent un set d'instructions basiques comme ajouter, supprimer ou récupérer des documents ou des fragments. De plus, elles possèdent des méthodes pour utiliser les indexes, comme par exemples récupérer un ensemble de nœuds correspondant à un certain nom. Les moteurs de requête Xpath et XQuery sont implémentes de la même manière, comme des modules gravitant autour du cœur d'eXist.

eXist propose plusieurs types de déploiements. Le moteur de base de données peut fonctionner comme un processus serveur autonome fournissant des interfaces http et XML-RPC[107] pour des accès déportés. Il peut être intégré à des applications, lesquelles peuvent avoir accès directement à la base de données via l'API `XML:DB`[108]. Enfin, il peut fonctionner à l'intérieur d'un serveur de servlet tel que Tomcat d'Apache. Les accès XML-RPC, SOAP[109] et WebDAV[110] sont fournis par les servlets

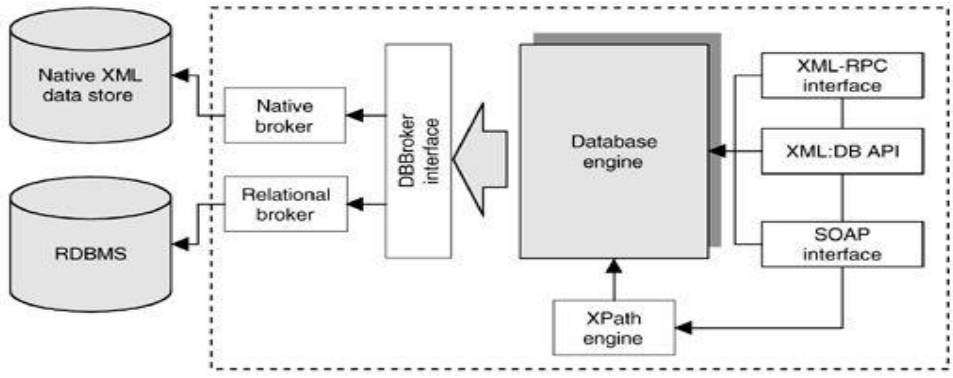


**Figure 53. Architecture d'eXist © Wolfgang Meier**

[107] http://xmlrpc.scripting.com/spec.html
[108] http://xmldb-org.sourceforge.net/xapi/xapi-draft.html
[109] http://www.w3.org/2000/xp/Group/
[110] http://www.ietf.org/rfc/rfc2518.txt

# BIBLIOGRAPHIE

# Bibliographie


Achard, F., Vaysseix, G. and Barillot, E. (2001) XML, bioinformatics and data integration, *Bioinformatics*, **17**, 115-125.

Aerts, K., Maesen, K. and Von Rompaey, A. (2006) A practical Example of Semantic Interoperability of Large-Scale Topographic Database using Semantic Web technologies. *9th AGILE International Conference on Geographic Information Science*. Visegrád, Hungary.

Alashqur, A.M., Su, S.Y.W. and Lam, H. (1989) OQL: A Query Language for Manipulating Object-oriented Databases. *Proceedings of the 15th International Conference on Very Large Data Bases (VLDB '89)*. Morgan Kaufmann, pp. 433–442.

Altschul, S.F.*, et al.* (1990) Basic local alignment search tool, *J Mol Biol*, **215** 403-410.

Arenson, A.D. (2003) Federating data with Information Integrator, *Briefings in Bioinformatics*, **4**, 375-381.

Ashburner, M.*, et al.* (2000) Gene ontology: tool for the unification of biology., *Nature genetics*, **25**, 25-29.

Ault, M.*, et al.* (2003) Oracle Database 10g New Features : Oracle10g Reference for Advanced Tuning and Administration. Rampant TechPress.

Baader, F.*, et al.* (2003) The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press,.

Baker, P.G.*, et al.* (1999) An ontology for bioinformatics applications, *Bioinformatics*, **15**, 510-520.

Balko, S.*, et al.* (2004) BioDataServer: an Applied Molecular Biological Data Integration Service Data Integration in the Life Sciences. In Rahm, E. (ed). Springer Berlin / Heidelberg, pp. 140-155.

Benitez-Guerrero, E., Collet, C. and Adiba, M. (1999) *Entrepôts de données : synthése et analyse*. Institut d'informatique et de mathématiques appliquées de Grenoble, Grenoble, FRANCE.

Benitez-Guerrero, E., Collet, C. and Adiba, M. (2001) Entrepôts de données : caractéristiques et problématique, *Technique et Science Informatiques*, **20**, 145 -178.

Benson, D.A.*, et al.* (2011) GenBank, *Nucleic Acids Research*, **39**, D32-D37.

Bernstein, P.A. and Rahm, E. (2000) Data warehouse scenarios for model management. *Proceedings of the 19th international conference on Conceptual modeling*. Springer-Verlag, Salt Lake City, Utah, USA, pp. 1-15.

Bilofsky, H.S. and Christian, B. (1988) The GenBank genetic sequence data bank, *Nucleic Acids Research*, **16**, 1861-1863.

Bishr, Y.A. (1998) overcoming the semantic and other barriers to gis interoperability, *International Journal of Geographical Information Science*, **12**, 299–314.

Blagosklonny, M.V. and Pardee, A.B. (2002) The Restriction Point of the Cell Cycle, *Cell Cycle*, **1**, 102-104.

Boguski, M.S., Lowe, T.M.J. and Tolstoshev, C.M. (1993) dbEST :database for [ldquo]expressed sequence tags[rdquo], *Nat Genet*, **4**, 332-333.

Boussaïd, O*., et al.* (2006) Conception et construction d'entrepôts en XML. *EDA'06*. Versaille.

Briache, A*., et al.* (2012) Transparent mediation-based access to multiple yeast data sources using an ontology driven interface, *BMC bioinformatics*, **13**, S7.

Brooksbank, C., Cameron, G. and Thornton, J. (2005) The European Bioinformatics Institute's data resources: towards systems biology, *Nucleic Acids Research*, **33**, D46-D53.

Brown, P.O. and Botstein, D. (1999) Exploring the new world of the genome with DNA microarrays, *Nat Genet*.

Buschmann, F*., et al.* (1996) Pattern-Oriented Software Architecture - A System of Patterns. John Wiley and Sons.

Calvanese, D*., et al.* (1998) Source Integration in Data Warehousing. *Proceedings of the 9th International Workshop on Database and Expert Systems Applications*. IEEE Computer Society, pp. 192.

Codd, E.F., Codd, S.B. and Salley, C.T. (1993) Providing OLAP (On-Line Analytical Processing) to User-Analysis: An IT Mandate. E. F. Codd & Associates.

Cohen-Boulakia, S., B., D.S. and Froidevaux, C. (2005) A User-Centric Framework for Accessing Biological Sources and Tools. *Data Integration in the Life Sciences*.

Cohen-Boulakia, S*., et al.* (2002) Genopage : A database of all protein modules encoded by completely sequenced genomes. *JOBIM 2002, Journees Ouvertes, Biologie, Informatique et Mathematiques*. pp. 187-193.

Cohen-Boulakia, S*., et al.* (2004) Selecting biomedical data sources according to user preferences, *Bioinformatics*, **20**, i86-i93.

Colonna, F.-M. (2008) Intégration de données hétérogènes et distribuées sur le Web et applications à la biologie. UNIVERSITÉ PAUL CÉZANNE AIX-MARSEILLE III.

Collaborative, T.P.G.D. (2001) PlasmoDB: An integrative database of the Plasmodium falciparum genome. Tools for accessing and analyzing finished and unfinished sequence data, *Nucleic Acids Research*, **29**, 66-69.

Committee, o.F.a.t.I.o.C.a.B. (2005) Catalyzing Inquiry at the Interface of Computing and Biology. National Research Council of the National Academies, Washington, Etats-Unis.

Consortium, T.U. (2010) The Universal Protein Resource (UniProt) in 2010, *Nucleic Acids Research*, **38**, D142-D148.

Cornell, M*., et al.* (2003) GIMS: an integrated data storage and analysis environment for genomic and functional data, *Yeast*, **20**, 1291-1306.

Chamberlin, D. (1998) A Complete Guide to DB2 Universal Database. Morgan Kaufmann, San Francisco, Californie.

Chang, A*., et al.* (2009) BRENDA, AMENDA and FRENDA the enzyme information system: new content and tools in 2009, *Nucleic Acids Research*, **37**, D588-D592.

Chaudhuri, S. and Dayal, U. (1997) An overview of data warehousing and OLAP technology, *SIGMOD Rec.*, **26**, 65-74.

Chen, R., Felciano, R. and Altman, R. (1997) RIBOWEB: Linking Structural Computations to a Knowledge Base of Published Experimental Data. *Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, pp. 84-87.

Chin-A-Woeng, T.F.C*., et al.* (2000) Root Colonization by Phenazine-1-Carboxamide-Producing Bacterium Pseudomonas chlororaphis PCL1391 Is Essential for Biocontrol of Tomato Foot and Root Rot, *Molecular Plant-Microbe Interactions*, **13**, 1340-1345.

Chin-A-Woeng, T.F.C*., et al.* (2001) Phenazine-1-Carboxamide Production in the Biocontrol Strain Pseudomonas chlororaphis PCL1391 Is Regulated by Multiple Factors Secreted into the Growth Medium, *Molecular Plant-Microbe Interactions*, **14**, 969-979.

Chniber, O. and Kerzazi, A., Navas-Delgado, I. and Aldana-Montes, J.F (2008) KOMF: The Khoas Ontology-based Mediator Framework. *NETTAB 2008: Bioinformatics Methods for Biomedical Complex System Applications*. Italy.

Choquet, R. and Boussaïd, O. (2007) Interrogation OLAP d'un entrepôt de données XML. *EGC'07 : Extraction et Gestion des Connaissances*. Belgique.

Davidson, S.B*., et al.* (2001) K2/Kleisli and GUS: experiments in integrated access to genomic data sources, *IBM Syst. J.*, **40**, 512-531.

Davidson, S.B., Overton, C. and Buneman, P. (1995) Challenges in integrating biological data sources, *Journal of Computational Biology*, **2**, 557–572.

Davidson, S.B*., et al.* (1997) BioKleisli: A Digital Library for Biomedical Researchers (1996), *Int. J. on Digital Libraries*, **1**, 36-53.

Do, H.-H. and Rahm, E. (2004) Flexible Integration of Molecular-biological Annotation Data: The GenMapper Approach. In E. Bertino, S.C., D. Plexousakis, V. Christophides, M. Koubarakis, K. Bohm, and E. Ferrari, (ed), *9th International Conference on Extending Database Technology*. Heraklion, Crete, Greece, pp. 811-822.

Donlin, M.J. (2002) Using the Generic Genome Browser (GBrowse). In, *Current Protocols in Bioinformatics*. John Wiley & Sons, Inc.

Ely, J.W*., et al.* (2000) A taxonomy of generic clinical questions: classification study, *British Medical Journal BMJ*, **321**, 429–432.

Emmanuel, B*., et al.* (2000) The taxonomy of Pseudomonas fluorescens and Pseudomonas putida: current status and need for revision, *Agronomie*, **20**.

Etzold, T. and Argos, P. (1993) SRS—an indexing and retrieval tool for flat file data libraries, *Computer applications in the biosciences : CABIOS*, **9**, 49-57.

Etzold, T., Ulyanov, A. and Argos, P. (1996) SRS: Information retrieval system for molecular biology data banks. In Russell, F.D. (ed), *Methods in Enzymology*. Academic Press, pp. 114-128.

Eyquem, A., Alouf, J. and Montagnier, L. (2005) Traité de microbiologie clinique. PICCIN, pp. 68.

Fasman, K.H., Cuticchia, A.J. and Kingsbury, D.T. (1994) The GDB Human Genome Data Base anno 1994., *Nucleic Acids Research*, **22**, 3462–3469.

Franco, J.-M. (1997) Le Data Warehouse - Le Data Mining. In Eyrolles (ed). Paris.

Friedman, M., Levy, A. and Millstein, T. (1999) Navigational plans for data integration. *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*. American Association for Artificial Intelligence, Orlando, Florida, United States, pp. 67-73.

Galperin, M.Y. and Fernández-Suárez, X.M. (2011) The 2012 Nucleic Acids Research Database Issue and the online Molecular Biology Database Collection, *Nucleic Acids Research*.

Galperin, M.Y. and Fernández-Suárez, X.M. (2012) The 2012 Nucleic Acids Research Database Issue and the online Molecular Biology Database Collection, *Nucleic Acids Research*, **40**, D1-D8.

Gasteiger, E.*, et al.* (2003) ExPASy: the proteomics server for in-depth protein knowledge and analysis, *Nucleic Acids Research*, **31**, 3784-3788.

Gautier, C. (1981) *Nucleic acid sequences handbook*. Praeger.

Glasner, J.D.*, et al.* (2008) Enteropathogen Resource Integration Center (ERIC): bioinformatics support for research on biodefense-relevant enterobacteria, *Nucleic Acids Research*, **36**, D519-D523.

Goble, C. (2002) Position Statement: Musings on Provenance, Workflow and (Semantic Web) Annotations for Bioinformatics. *DansWorkshop on Data Derivation and Provenance*.

Griffith, A. (2005) Java, XML, and the JAXP. In Wiley (ed).

Gruber, T.R. (1995) Toward principles for the design of ontologies used for knowledge sharing, *Int. J. Hum.-Comput. Stud.*, **43**, 907-928.

Guérin, E.*, et al.* (2005) Integrating and warehousing liver gene expression data and related biomedical resources in GEDAW. *Proceedings of the Second international conference on Data Integration in the Life Sciences*. Springer-Verlag, San Diego, CA, pp. 158-174.

Gupta, P. and Lin, E. (1994) DataJoiner: a practical approach to multi-database access. *Parallel and Distributed Information Systems, 1994., Proceedings of the Third International Conference on*. pp. 264.

Haas, D. and Keel, C. (2003) REGULATION OF ANTIBIOTIC PRODUCTION IN ROOT-COLONIZING PSEUDOMONAS SPP. AND RELEVANCE FOR BIOLOGICAL CONTROL OF PLANT DISEASE, *Annual Review of Phytopathology*, **41**, 117-153.

Haas, L.M.*, et al.* (2001) DiscoveryLink: A system for integrated access to life sciences data sources, *IBM Systems Journal*, **40**, 489-511.

Hamm, G.H. and Cameron, G.N. (1986) The EMBL data library, *Nucleic Acids Research*, **14**, 5-9.

Hammer J and Schneider M ( 2003) Going back to our database roots for managing genomic data, *OMICS.*, **7**, 117-119.

Harold, E.R. and Means, W.S. (2004) XML in a Nutshell. O'Reilly Media.

Hart, K.*, et al.* (1994) Using a Query Language to Integrate Biological Data. *1st meeting on the Interconnection of Molecular Biology Databases*. Stanford, California, USA.

Hartmann, J.*, et al.* (2005) Ontology Metadata Vocabulary and Applications On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops. In Meersman, R., Tari, Z. and Herrero, P. (eds). Springer Berlin / Heidelberg, pp. 906-915.

Hernandez, T. and Kambhampati, S. (2004) Integration of biological sources: current systems and challenges ahead, *SIGMOD Rec.*, **33**, 51-60.

Hillebrand, G.G.*, et al.* (1995) Undecidable Boundedness Problems for Datalog Programs, *J. of Logic Programming*, **25**, 163--190.

Hood, L. and Galas, D. (2003) The digital code of DNA, *Nature*, **421**, 444-448.

Hunter, J. (2003) X is for Query. *Oracle Magazine*.

Inmon, W.H. (1996) Building the data warehouse. In Wiley, J., Sons and Sons (eds). New York.

Inmon, W.H. (2002) Building the Data Warehouse. In Wiley, J. (ed).

Jagadish, H.V., Lakshmanan, L.V.S. and Srivastava, D. (1999) What can Hierarchies do for Data Warehouses? , *Proceedings of the 25th International Conference on Very Large Data Bases*. Morgan Kaufmann Publishers Inc., pp. 530-541.

Jagadish, H.V. and Olken, F. (2003) Data Management for the Biosciences, Report of the NSF/NLM Workshop on Data Management for Molecular and Cell Biology.

Kadima, H. and Monfor, V. (2003) Les Web Services : techniques, d´emarches et outils. In DUNOD (ed).

Kanehisa, M. and Goto, S. (2000) KEGG: Kyoto Encyclopedia of Genes and Genomes, *Nucleic Acids Research*, **28**, 27-30.

Kanehisa, M.*, et al.* (2006) From genomics to chemical genomics: new developments in KEGG, *Nucleic Acids Research*, **34**, D354-D357.

Kanehisa, M.*, et al.* (2004) The KEGG resource for deciphering the genome, *Nucleic Acids Research*, **32**, D277-D280.

Karp, P.D.*, et al.* (2000) The EcoCyc and MetaCyc databases, *Nucleic Acids Research*, **28**, 56-59.

Kasprzyk, A.*, et al.* (2004) EnsMart: A Generic System for Fast and Flexible Access to Biological Data, *Genome Research*, **14**, 160-169.

Katz, H.*, et al.* (2003) Xquery from the Experts: A Guide to the W3C Xml Query Language. Addison Wesley.

Keseler, I.M.*, et al.* (2005) EcoCyc: a comprehensive database resource for Escherichia coli, *Nucleic Acids Research*, **33**, D334-D337.

Kimball, R. (2002) data warehouse toolkit.

Kimball, R. (2003) The Bottom-Up Misnomer.

King, R.A., Hameurlain, A. and Morvan, F. (2008) Ontology-based data source localization in a structured peer-to-peer environment. *Proceedings of the 2008 international symposium on Database engineering \&\#38; applications*. ACM, Coimbra, Portugal, pp. 9-18.

Kirsten, T., Do, H.-H.D. and Rahm, E. (2004) A Data Warehouse for Multidimensional Gene Expression Analysis. *Technical Report, IZBI Working Paper*.

Lacot, X. (2005) Introduction à OWL, un langage XML d'ontologies Web

Lacroix, Z. and Edupuganti, V. (2004) How biological source capabilities may affect the data collection process. *Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004 IEEE*. pp. 596-597.

Lacroix, Z.*, et al.* (2005a) BioNavigation: selecting optimum paths through biological resources to evaluate ontological navigational queries. *Proceedings of the Second international conference on Data Integration in the Life Sciences*. Springer-Verlag, San Diego, CA, pp. 275-283.

Lacroix, Z.*, et al.* (2005b) BioNavigation: using ontologies to express meaningful navigational queries over biological resources. *Computational Systems Bioinformatics Conference, 2005. Workshops and Poster Abstracts. IEEE*. pp. 137-138.

Lans, R.F.V.D. (1989) The SQL standard : a complete guide reference. Prentice Hall International Ltd., Hertfordshire, Royaume-Uni.

Lee, T.*, et al.* (2006) BioWarehouse: a bioinformatics database warehouse toolkit, *BMC bioinformatics*, **7**, 170.

Levy, A.Y. (1999) Combining artificial intelligence and databases for data integration. In Michael, J.W. and Manuela, V. (eds), *Artificial intelligence today*. Springer-Verlag, pp. 249-268.

Lipman, D.J. and Pearson, W.R. (1985) Rapid and sensitive protein similarity searches, *Science*, **227**, 1435–1441.

List, B.*, et al.* (2002) A Comparison of Data Warehouse Development Methodologies Case Study of the Process Warehouse Database and Expert Systems Applications. In Hameurlain, A., Cicchetti, R. and Traunmüller, R. (eds). Springer Berlin / Heidelberg, pp. 203-215.

MacGregor R and Bates R (1987) The Loom knowledge representation language. *ISI/RS-87-188*. University of Southern California, Information Science Institute, Marina del Rey, CA.

Mahboubi, H.*, et al.* (2009) Enhancing XML data warehouse query performance by fragmentation. *Proceedings of the 2009 ACM symposium on Applied Computing*. ACM, Honolulu, Hawaii, pp. 1555-1562.

Mahoui, M.*, et al.* (2005) Semantic correspondence in federated life science data integration systems. *Proceedings of the Second international conference on Data Integration in the Life Sciences*. Springer-Verlag, San Diego, CA, pp. 137-144.

Markowitz, V.M.*, et al.* (2005) The integrated microbial genomes (IMG) system, *Nucleic Acids Research*, **34**, D344-D348.

Marrakchi, K.*, et al.* (2010) A Data Warehouse Approach to Semantic Integration of Pseudomonas Data, Data Integration in the Life Sciences,. In Lambrix, P. and Kemp, G. (eds). Springer Berlin / Heidelberg, pp. 90-105.

Martin, D.W.*, et al.* (1993) Mechanism of conversion to mucoidy in Pseudomonas aeruginosa infecting cystic fibrosis patients, *Proceedings of the National Academy of Sciences*, **90**, 8377-8381.

Martin, P. (1996) Exploitation de graphes conceptuels et de documents structurés et hypertextes pour l'acquisition de connaissances et la recherche d'informations. pp. 378.

Mazzarelli, J.M.*, et al.* (2007) EPConDB: a web resource for gene expression related to pancreatic development, beta-cell function and diabetes, *Nucleic Acids Research*, **35**, D751-D755.

McLaughlin, B. (2002) Java & XML Data Binding. In Media, O.R. (ed).

McLeod, M.P.*, et al.* (2006) The complete genome of Rhodococcus sp. RHA1 provides insights into a catabolic powerhouse, *Proceedings of the National Academy of Sciences*, **103**, 15582-15587.

Mewes, H.W.*, et al.* (2002) MIPS: a database for genomes and protein sequences, *Nucleic Acids Research*, **30**, 31-34.

Minoru, K. (1997) A database for post-genome analysis, *Trends in Genetics*, **13**, 375-376.

Mork, P., Halevy, A. and Tarczy-Hornoch, P. (2001) A model for data integration systems of biomedical data applied to online genetic databases. *Proc AMIA Symp*. pp. 473–477.

Mork, P., Halevy, A. and Tarczy-Hornoch, P. (2002) PQL: a declarative query language over dynamic biological schemata. *Proc AMIA Symp*. pp. 533-537.

Morris, S.B. (2003) Network Management, MIBs and MPLS: Principles, Design and Implementation. Prentice Hall.

Moszer, I.*, et al.* (2002) SubtiList: the reference database for the Bacillus subtilis genome, *Nucleic Acids Research*, **30**, 62-65.

Münch, R.*, et al.* (2003) PRODORIC: prokaryotic database of gene regulation, *Nucleic Acids Research*, **31**, 266-269.

Navas-Delgado, I. (2008) An Infrastructure for Developing Applications in the Semantic Web. UNIVERSIDAD DE MALAGA Higher Technical School of Computer Science Engineering, Malaga.

Navas-Delgado, I. and Aldana-Montes, J. (2008) SD-Core: Generic Semantic Middleware Components for the Semantic Web Knowledge-Based Intelligent Information and Engineering Systems. In Lovrek, I., Howlett, R. and Jain, L. (eds). Springer Berlin / Heidelberg, pp. 617-622.

Navas-Delgado, I. and Aldana-Montes, J.F. (2009) Extending SD-Core for Ontology-based Data Integration. , *J.UCS*, **15**, 3201-3230.

Olken, F. and Jagadish, H.V. (2003) Data Management for Integrative Biology, *OMICS*, **7**, 1-2.

Pandey, A. and Mann, M. (2000) Proteomics to study genes and genomes, *Nature*, **405**, 837-846.

Peterson, J.D.*, et al.* (2001) The Comprehensive Microbial Resource, *Nucleic Acids Research*, **29**, 123-125.

Rahm, E. and Bernstein, P.A. (2001) A survey of approaches to automatic schema matching, *The VLDB Journal*, **10**, 334-350.

Rebhan, M.*, et al.* (1997) GeneCards: integrating information about genes, proteins and diseases, *Trends in Genetics*, **13**, 163.

Rector, A.L.*, et al.* (1997) The GRAIL concept modelling language for medical terminology, *Artificial Intelligence in Medicine*, **9**, 139-171.

Reese, G. (2001) JDBC et Java - Guide du programmeur. In O'Reilly (ed).

Rehm, B. (2009) Pseudomonas. Wiley-VCH.

Roth, M.T.*, et al.* (1996) The Garlic project, *SIGMOD Rec.*, **25**, 557.

Roychoudhury, S.*, et al.* (1992) Characterization of guanosine diphospho-D-mannose dehydrogenase from Pseudomonas aeruginosa. Structural analysis by limited proteolysis, *Journal of Biological Chemistry*, **267**, 990-996.

Schöning, D.H. (2001) Tamino - A DBMS Designed for XML. *Proceedings of the 17th International Conference on Data Engineering*. IEEE Computer Society, pp. 149.

Sen, A. and Sinha, A.P. (2005) A comparison of data warehousing methodologies, *Commun. ACM*, **48**, 79-84.

Sen, T.Z.*, et al.* (2010) Choosing a genome browser for a Model Organism Database: surveying the Maize community, *Database*, **2010**.

Shaker, R.*, et al.* (2002) Rule Driven Bi-Directional Translation System Remapping Queries and Result Sets Between a Mediated Schema and Heterogeneous Data Sources. *Proc AMIA Symp*. American Medical Informatics Association, pp. 692-696.

Sheth, A.P. and Larson, J.A. (1990) Federated database systems for managing distributed, heterogeneous, and autonomous databases, *ACM Comput. Surv.*, **22**, 183-236.

Shin, D., Jang, H. and Jin, H. (1998) BUS: an effective indexing and retrieval scheme in structured documents. *Proceedings of the third ACM conference on Digital libraries*. ACM, Pittsburgh, Pennsylvania, United States, pp. 235-243.

Sidman, K.E.*, et al.* (1988) The protein identification resource (PIR), *Nucleic Acids Research*, **16**, 1869-1871.

Stephens, J. and Russell, C. ( 2004) Beginning MySQL Database Design and Optimization. Springer-Verlag, New York.

Stevens, R.*, et al.* (2000) TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources, *Bioinformatics*, **16**, 184-186.

Stevens, R.*, et al.* (2001) A classification of tasks in bioinformatics, *Bioinformatics*, **17**, 180-188.

Stevens, R.*, et al.* (2002) Building a bioinformatics ontology using OIL, *Information Technology in Biomedicine, IEEE Transactions on*, **6**, 135-141.

Sujansky, W. (2001) Heterogeneous database integration in biomedicine, *Comput. Biomed. Res.*, **34**, 285-298.

Sun, W. and Liu, D.-X. (2006) Using Ontologies for Semantic Query Optimization of XML Database Knowledge Discovery from XML Documents. In Nayak, R. and Zaki, M. (eds). Springer Berlin / Heidelberg, pp. 64-73.

Thomas, J. and Stefan, D. (2008) Towards generating ETL processes for incremental loading. *Proceedings of the 2008 international symposium on Database engineering applications*. ACM, Coimbra, Portugal, pp. 101-110.

Toumani, K., Jaudoin, H. and Schneider, M. (2007) Génération automatique de correspondances sémantiques entre schémas. *INFORSID*. pp. 261-276.

Walter, S. (2001) Heterogeneous Database Integration in Biomedicine, *Journal of Biomedical Informatics*, **34**, 285-298.

Wall, L. (2000) Programming Perl. O'Reilly & Associates, Sebastopol, Californie, Etats-Unis,.

Waugh, A.*, et al.* (2002) RNAML: a standard syntax for exchanging RNA information, *RNA*, **8**, 707-717.

Wiederhold, G. (1992) Mediators in the Architecture of Future Information Systems, *Computer*, **25**, 38-49.

Winsor, G.L.*, et al.* (2009) Pseudomonas Genome Database: facilitating user-friendly, comprehensive comparisons of microbial genomes, *Nucleic Acids Research*, **37**, D483-D488.

Xuan, W.*, et al.* (2009) Open Biomedical Ontology-based Medline exploration, *BMC bioinformatics*, **10**, S6.

Zdobnov, E.M.*, et al.* (2002) The EBI SRS server—new features, *Bioinformatics*, **18**, 1149-1150.

Zdobnov, E.M.*, et al.* (2002) The EBI SRS server—recent developments, *Bioinformatics*, **18**, 368-373.

Zimmermann, R.*, et al.* (2006) A Distributed Geotechnical Information Management and Exchange Architecture, *Internet Computing, IEEE*, **10**, 26-33.

# Références Internet

# Références Internet

| | |
|---|---|
| (NCBI), "Microbial Genomes" | http://www.ncbi.nlm.nih.gov/genomes/MICROBES/microbial_taxtree.html |
| AmiGO | http://amigo.geneontology.org/cgi-bin/amigo/go.cgi |
| Apache Server | http://httpd.apache.org/ |
| ArrayExpress | http://www.ebi.ac.uk/arrayexpress/ |
| ASN | http://www.bgbm.org/tdwg/acc/Documents/asn1gloss.htm |
| Auto-formation en Bioinformatique | http://www.dsi.univ-paris5.fr/bio2/autof2/cha2_int.htm |
| Axis | http://ws.apache.org/axis/overview.html |
| BioCyc | http://biocyc.org/ |
| BioGrid | http://thebiogrid.org/ |
| Bioperl | http://www.bioperl.org/wiki/Main_Page |
| biosql | http://www.biosql.org/wiki/Main_Page |
| Blast | http://blast.ncbi.nlm.nih.gov/Blast.cgi |
| Bots | http://en.wikipedia.org/wiki/Wikipedia:Bots |
| BRENDA | http://www.brenda-enzymes.info/ |
| Chado | http://gmod.org/wiki/Chado_-_Getting_Started |
| ChEBI | http://www.ebi.ac.uk/chebi/ |
| CMR | http://cmr.jcvi.org/tigr-scripts/CMR/CmrHomePage.cgi |
| core | http://dublincore.org/ |
| CYGD-MIPS | http://mips.helmholtz-muenchen.de/genre/proj/yeast/ |
| dbEST | http://www.ncbi.nlm.nih.gov/dbEST/ |
| dbSNP | http://www.ncbi.nlm.nih.gov/projects/SNP/ |
| DDBJ | http://www.ddbj.nig.ac.jp/ |
| Dublin Core | http://dublincore.org/ |
| EBI | http://www.ebi.ac.uk/ |
| EcoCyc | http://ecocyc.org/ |
| EMBL | http://www.embl.de/ |
| EMBO | http://www.embo.org/ |
| ensEMBL | http://www.ensembl.org/index.html |
| Enteropathogen Resource Integration Center | http://patricbrc.vbi.vt.edu/portal/portal/patric/IncumbentBRCs?page=eric |
| *Entrez* | http://www.ncbi.nlm.nih.gov/sites/gquery |
| EPConDB | http://www.cbil.upenn.edu/epcondb42/ |
| eXist | http://exist.sourceforge.net |

| | |
|---|---|
| *ExPASy* | http://expasy.org/ |
| ExPASy | http://expasy.org/ |
| Extension_Matrix | http://www.mediawiki.org/wiki/Extension_Matrix |
| FASTA | http://www.ebi.ac.uk/Tools/sss/fasta/ |
| Flybase | http://flybase.org/ |
| Garlic | http://www.almaden.ibm.com/cs/garlic/ |
| Gbrowse | http://gmod.org/wiki/GBrowse |
| GDB | http://gdbwww.gdb.org/ |
| Genbank | http://www.ncbi.nlm.nih.gov/nuccore/ |
| GeneCards | http://www.genecards.org/ |
| GenMapper | http://ducati.izbi.uni-leipzig.de:8080/GenMapper/servlet/gui.MainFrame |
| GEO | http://www.ncbi.nlm.nih.gov/geo/ |
| GeWare | http://ducati.izbi.uni-leipzig.de:8080/Geware/servlet/de.izbi.geware.common.forms.FrameSet |
| GFF | http://gmod.org/wiki/GFF |
| GO | http://www.geneontology.org/ |
| HGNC | http://www.genenames.org/ |
| IMG | http://img.jgi.doe.gov |
| inmon | http://en.wikipedia.org/wiki/Bill_Inmon |
| InterPro | http://www.ebi.ac.uk/interpro/ |
| Java DOM | http://docs.oracle.com/javase/1.4.2/docs/api/org/w3c/dom/package-summary.html |
| JCVI CMR | http://cmr.jcvi.org/tigr-scripts/CMR/CmrHomePage.cgi |
| jena | http://jena.apache.org/ |
| Jetty | http://jetty.codehaus.org/jetty/ |
| JWBF | http://jwbf.sourceforge.net/ |
| KEGG | http://www.genome.jp/kegg/ |
| LION Bioscience AG | http://www.biochipnet.com/node/1561 |
| MediaWiki configuration | http://www.mediawiki.org/wiki/Category:MediaWiki_configuration_settings |
| Medline | http://www.medline.com/ |
| MeSH | http://www.nlm.nih.gov/mesh/ |
| MetaCyc | http://metacyc.org/ |
| MGI | http://www.informatics.jax.org/ |
| Microbes Online | http://www.microbesonline.org/ |
| MIPS | http://www.helmholtz-muenchen.de/en/ibis |
| MySQL | http://www.mysql.com/ |
| NCBI | http://www.ncbi.nlm.nih.gov/ |
| NIH | http://www.nih.gov/ |
| OBO | http://www.obofoundry.org/ |
| ODMG | www.odmg.org |
| OMIM | http://www.omim.org/ |
| ORACLE | http://www.oracle.com/index.html |
| OWL | http://www.w3.org/TR/2009/WD-owl2-primer-20090611/ |
| PDB | http://www.rcsb.org/pdb/home/home.do |

| | |
|---|---|
| peer-review literature | http://en.wikipedia.org/wiki/Peer_review |
| perl | http://dev.perl.org/perl5/ |
| Pfam | http://pfam.sanger.ac.uk/ |
| PhosphGrid | http://www.phosphogrid.org/ |
| Plasmodb | http://plasmodb.org/plasmo/ |
| ProDom | http://prodom.prabi.fr/prodom/current/html/home.php |
| PRODORIC | http://www.prodoric.de/ |
| Protégé | http://protege.stanford.edu/ |
| Pseudomonas Genome Database | http://www.pseudomonas.com/ |
| Pseudomonas syringae Genome Resources | http://www.pseudomonas-syringae.org/ |
| PseudomonasDW | http://www.pseudomonasdw.khaos.uma.es |
| PubMed | http://www.ncbi.nlm.nih.gov/pubmed/ |
| Qexo | http://www.xml.com/pub/a/2003/06/11/qexo.html |
| RDF | http://www.w3.org/TR/rdf-concepts/ |
| RDFS | http://www.w3.org/TR/rdf-schema/ |
| RefSeq | http://www.ncbi.nlm.nih.gov/RefSeq/ |
| RiboWeb | http://helix-web.stanford.edu/riboweb.html |
| SGD database | http://www.yeastgenome.org |
| SRS | http://srs.ebi.ac.uk/ |
| Tomcat | http://tomcat.apache.org/ |
| UML | http://www.uml.org/ |
| UMLS | http://www.nlm.nih.gov/research/umls/ |
| UniGene | http://www.ncbi.nlm.nih.gov/unigene |
| UniProt | http://www.uniprot.org/ |
| W3C | http://www.w3.org/ |
| watchlist | http://www.mediawiki.org/wiki/Manual:Watchlist |
| WebDAV | http://www.ietf.org/rfc/rfc2518.txt |
| Wikipedia | http://www.wikipedia.org/ |
| xBASE | http://www.xbase.ac.uk/ |
| XML | http://www.w3schools.com/xml/ |
| XML :DB | http://xmldb-org.sourceforge.net/xapi/xapi-draft.html |
| XML-RPC | http://xmlrpc.scripting.com/spec.html |
| XML-RPC, SOAP | http://www.w3.org/2000/xp/Group/ |
| ZFIN | http://zfin.org/ |