
XMLpipeDB:
Relational Database Schemas
Piped From
XML Descriptions
In Near-real Time

User's Manual
Tutorial Documentation
Release Instructions



BIOL 498 / CMSI 698
Loyola Marymount University
Spring Semester, 2006

XMLPIPEDB:

RDB SCHEMAS PIPED FROM XML DESCRIPTIONS

SPECIAL STUDIES: BIOINFORMATICS
BIOL 498 / CMSI 698 – SPRING 2006

KAM DAHLQUIST, PHD., JOHN DIONISIO, PHD.,
J. BARRETT, J. BOYLE, A. CARASSO, D. HOFFMAN, B. JOHNSON, B. NAFFAS, J. NICHOLAS, R. RUIZ, S. SPICER
LOYOLA MARYMOUNT UNIVERSITY

1. Introduction

Since the completion of the human genome sequencing effort, the amount of data that is available to researchers has skyrocketed. Not only has the number of data banks containing the information increased, but the amount of data that is stored in those databases has grown exponentially. There are now databanks containing information for all facets of gene investigation, protein sequencing, viral investigations, and many other topics. The sheer volume of data, along with the sophisticated tools required to manipulate, store, retrieve, study, and understand that data, requires extensive application of technology. Hence, the emergence of *Bioinformatics*.

Bioinformatics is the application of Information Technology to biological data. This information technology provides the ability to represent, organize, manipulate, distribute, maintain, and use the data. It is an interdisciplinary scientific endeavor, comprised primarily of researchers in biology and computer science, but incorporating mathematics, chemistry, physics, engineering, and other disciplines as well.

The ultimate goal of Bioinformatics is to allow scientists to predict the traits of an organism and the organism's response to its environment based only on its genome sequence. But there is a problem, namely, the veritable deluge of data being produced. How does one begin to analyze data from thousands of genes, taken over multiple time points, that are stored in dozens of data banks all over the world?

1.1. The Problem of Storage

The obvious solution, and the realistic application is the use of the Internet. Almost every databank has internet connectivity, allowing users to access and download information. Most have methods to upload new information to add to the knowledge base, with different methods of verification for this data. This causes a number of problems.

First, there is the difference between the underlying structure, or *schema*, of the database itself. Although a certain amount of unification of structure has been achieved, owing to de-facto standardization, there is still a great amount of variability.

Second, there are problems created by the normal updating process of the stored data in these databases. Since new data is made available frequently, it is possible that previously downloaded data may be obsolete, or even proved to be incorrect. While frequent updating is good for the databank that contains all the latest data, it is difficult for its subscribers to keep up. In addition, these updates may require a data bank to modify its format or its schema, to change locations to a different Web address altogether, or even to disappear completely.

Third, there is significance to the effect of having different methods of verification of the data itself, a process known as *curation*. Some data obtained experimentally, while other data are obtained from other data. The validation of these data, whether done by hand or by machine, is critical to the correctness of the data. The computer science principle of "garbage in = garbage out" applies heavily here.

Finally, there are issues of interoperability between different database engines and different operating system platforms, as well as issues with funding of the projects involved.

These problems add several layers of complexity to an already complex problem. We believe the researcher should be able to concentrate on her research, rather than having to be bogged down with the "computer science" of the underlying computer application tools she is using.

1.2. The Unifying Solution

Modern internet technology can solve these problems by allowing the researcher to install and maintain copies of the data of interest on local computers

Demonstrate how important it is that research be performed on the data without having to worry about the underlying application. Show how our application allows the researcher to focus on the information rather than the application.

1.3. Document Organization

The remainder of this document is organized as follows. Section 2 contains a tutorial about the XMLpipeDB processing. Section 3 contains a User's Guide, including all installation instructions and an example database translation case. Section 4 presents contact information, links to data banks, and other supporting information.

2. Tutorial

The following is a brief tutorial on the XMLpipeDB translation process, and how to use the program to create a translated database. We will then describe the process of how to translate and store data, and how to access that stored data.

2.1. XMLpipeDB Translation Process

There are three phases of the translation process; database definition, database loading, and data retrieval. Database definition consists of the steps of describing the remote data, translating that description, and creating the local database. Database loading consists of retrieving the remote data, then using the translating description to load the data into the local database. Database retrieval comes into play when recovering the stored data from the local database. The following sections describe both phases of this process.

2.2. Using XMLpipeDB to Create a Database

The database definition process begins with finding the data required for your research. This might be contained in any of the large number of genomic databanks currently available. Many of these repositories include links to files that describe the database in a particular format known as an "XSD" file. XSD stands for

2.3. Using XMLpipeDB to Translate and Store Data

Blahblahblah

2.4. Accessing the Stored Data

Blahblahblah

3. Example Database Translation

This section provides an example using a databank entry from the UniProt Protein Data Bank.

3.1. Assumptions / Requirements

3.2. Obtaining Required Files

There are a number

3.3. Installation of the local Database Engine

3.4. Installation of the Eclipse Integrated Development Environment

3.5. Installation and Configuration of the XMLpipeDB Files

3.6. Downloading of the data bank database description (XSD file)

3.7. Processing the XSD File

3.8. Downloading and Processing the Remote Data

3.9. Accessing the Local data

3.10. Customizing the Application

4. Support Information

The following URL links, papers, and books provide more information about Bioinformatics and the background of the XMLpipeDB, along with links to many of the major data banks for which this project is initially intended.

4.1. Internet Links (URLs)

NCBI: National Center for Biotechnology Information:

GenBank: DNA and protein sequence (contains same info as EMBL and DDBJ)

<http://www.ncbi.nlm.nih.gov/Genbank/index.html>

RefSeq: curated records from the above

<http://www.ncbi.nlm.nih.gov/RefSeq/>

UniGene: expressed sequence tags (ESTs)

<http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=unigene>

Gene: one stop shop for information about a particular gene/protein

<http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene>

PubMed: biomedical literature

<http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?DB=pubmed>

OMIM/OMIA: Online Mendelian Inheritance in Man

<http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=OMIM>

Taxonomy: hierarchy of organism nomenclature

<http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=OMIM>

BLAST: not a database, but useful tool

<http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/information3.html>

UniProt (formerly SwissProt): hand-curated protein sequence database

<http://www.uniprot.org>

TrEMBL: translated EMBL, not curated

<http://www.expasy.org/sprot/>

PDB: Protein Structure Database

<http://www.rcsb.org/pdb/Welcome.do>

Pfam: protein families

<http://www.sanger.ac.uk/Software/Pfam/>

Gene Ontology (GO): structured, controlled vocabularies and classifications widely used in genome annotation

<http://www.geneontology.org/>

Model Organism Databases

Saccharomyces Genome Database (SGD)

<http://www.yeastgenome.org/>

Mouse Genome Informatics (MGI)

<http://www.informatics.jax.org/>

WormBase

<http://www.wormbase.org/>

FlyBase

<http://flybase.bio.indiana.edu/>

TAIR (The *Arabidopsis* Information Resource)

<http://www.arabidopsis.org/>

TIGR (The Institute for Genome Research) hundreds of microbial genomes

<http://www.tigr.org/>

Human Genome Browser (UC Santa Cruz)

<http://genome.ucsc.edu/>

KEGG: metabolic pathways

<http://www.genome.jp/kegg/>

BioCyc: family of metabolic pathway databases

<http://www.biocyc.org/>

EcoCyc

<http://ecocyc.org/>

HumanCyc

<http://humancyc.org/>

4.2. Books

4.3. Interesting Papers