Integrating Biological Data and Tools with BIS

Zoé Lacroix, Omar Boucelma and Mehdi Essid

Abstract — The access and exploitation of integrated data repositories and applications is critical for life science. Biologists design protocols that typically rely on complex query pipelines accessing various biological electronic resources (data sources and tools) to constitute data sets for analysis and mining. Integration platforms are needed to allow biologists to access, manipulate and analyze electronic biological data and thus to support scientific discovery. The design of integration architectures to support life science addresses specific issues. Biological resources are highly heterogeneous: not only they differ by their data representation, but they also offer radically different query capabilities. In this paper we present the Biological Integration System (BIS) that focuses on the data integration aspects while addressing the integration of query capabilities available at the sources. Typically, biological data sources provide complex query and analysis capabilities such as sequence similarity search engines, that need to be integrated as well as the data. We introduce the notion of derived wrappers that capture additional capabilities to either compensate capabilities lacking at a source, or to adjust an existing capability in order to make it homogeneous with similar capabilities at other sources. BIS combines a data-driven approach with its mediator with an application-driven approach exploiting Web services.

Index Terms — Bioinformatics, Mediation, Heterogeneous Resources, Data Integration, Application Integration, Web.

I. INTRODUCTION

Scientists today spend significant time and effort in querying multiple remote or local heterogeneous data sources, and integrating the results, either manually, or with the aid of data integration tools, so that they may be further manipulated using advanced data analysis and visualization tools. Biological resources are either publicly available on the Web, or local and private. They include thousands of public data sources: there
are over 337 relevant resources in molecular biology only [1]. Each data source contains a large amount of data: GenBank [2] provides access to 10,336,000,000 bases in 9,103,000 sequences [3], human proteins (estimated to number about 100,000) will be available online [4]. Biological data is available in a wide variety of formats, annotated, and stored in flat files and relational, object-relational or object-oriented databases. Access to heterogeneous biological data sources is mandatory to life scientists. A single query may involve flat files (that may be stored locally) such as GenBank [2] or SWISS-PROT [5], Web resources such as GeneCards [6], or the references data source PubMed [7]. Biological resources (and in particular the ones publicly available on the Web) do not provide users with a real expressive query language such as SQL to access the data they contain. Instead, they provide a wide range of useful tools such as text search engines or similarity search tools such as FASTA [8], BLAST [9] or LASSAP [10], [11] that generate additional information needed to access the data. Getting access to biological resources, and combining data from, multiple data sources, while coping with their distribution and heterogeneity, is a tremendously difficult task.

Past efforts to support life scientists in their needs to exploit data and applications were developed in two orthogonal directions: data-driven and application-driven. Architectures for data integration have been extensively investigated [12]. Data warehouses and middleware solutions were developed to facilitate interoperability and data exchange, while heterogeneous distributed database management systems and mediation techniques enable data integration. These approaches are data-driven and rely on wrappers [13], [14], [15], [16], [17] to access data sources and to retrieve and translate the results into some common integrated representation. Data warehouses often use wrappers to import data from remote sources that are then materialized locally, whereas queries are evaluated locally against the warehoused data. A key disad-
vantage of the warehouse approach is the need for local administrators to maintain the data, while a key advantage is the control that it provides over the contents of the warehoused data. Mediators and heterogeneous distributed database management systems, on the other hand, submit queries to wrappers, and integrate the results locally to provide answers to queries. Accessed data is up-to-date, but data access can be costly. In addition, wrappers must be maintained for data providers may change frequently the entry points to the data sources as well as the data organization (schema).

Both materialized into data warehouses, or non-materialized approaches developed do not address the problem to access or maintain sophisticated tools that enhance data manipulation. Indeed, biological queries involve various applications in addition to standard data manipulation such as expressed by SQL. Biological data sources such as PubMed or GenBank typically provide search capabilities and links to other sources. Search capabilities include textual search, sequence similarity search such as BLAST, etc. Links are expressed as hyperlinks between data sources providing users with the browsing capability to click corresponding sequence in GenBank, for example. These query capabilities may be available partially or totally at some of the data sources. Similar operators may not be semantically equivalent at two different sources, or even at a given source! A biological data warehouse must provide an expressive query language that may be difficult to maintain and may be less expressive than the ones available at the remote sources. On the other hand, non-materialized approaches must not only integrate data, but also all these query capabilities of interest. The approach presented in the paper is a mediation approach that integrates both data and source capabilities.

Most of the existing mediation approaches such as TSIMMIS [18], HERMES [19], Information Manifold [20], or DISCO [21] focus on
the data integration aspects without providing any facility to integrate available tools. Other approaches have been developed to integrate scientific data such as biomolecular data [22]. They include BioKleisli [23], [24] and its extensions K2 [25] and Pizzkell/Kleisli [26], the multi-database system based on the Object Protocol Model (OPM) [27] and its Object-Web Wrapper (OWW) [17], DiscoveryLink [28], extension for life science of the DataJoiner based upon DB2 [29] merged with Garlic [13], P/FDM [30], [31] and TAMBIS [32]. Most of these approaches devoted to biological data integration wrapped some of the available query capabilities into specific data classes for OPM or user functions for DiscoveryLink. The OWW made it easy to maintain the access to query capabilities by representing them into an intermediate view mechanism called search views. None of the above approach would enable the homogenization or extension of the source capabilities into a rich query language available to the user.

In contrast, other approaches focus on the integration of applications with agent architectures (CORBA), Web services and the GRID. Examples of application-driven efforts to integrating biological resources include [33], [30], [34], [35] and various biological grids such as myGRID. Unlike the data-driven approaches presented above, they offer flexible access to needed applications: new applications can easily be added, existing applications can be updated. However, their framework for integrating distributed resources relies on shared APIs and resource descriptions with little focus on data management. The recent european effort DataGrid is aiming to setting up a computational and data-intensive grid of biological resources. These efforts should provide the necessary framework to permit the integration of resources distributed over the Internet without resolving various data management problems such as query evaluation, optimization, and transactions.

The approach presented in the paper is exploiting the two orthogonal efforts: it offers a mediation architecture to provide tra-
ditional data management, as well as wrappers to access various applications, Web services, and the GRID. The Biological Integration System (BIS) uses standard wrappers to access data, extended by derived wrappers that capture additional query capabilities. This approach offers several advantages. The first contribution is the ability to integrate schemas through Inter-schema Correspondence Assertions (ICA) that resolve heterogeneous schema mapping issues. The second contribution of the presented approach is the notion of derived wrappers that can either capture query capabilities available at the source or access a local query capability not available at the source in order to make it available to the user. This approach enables a rich query language in contrast to an approach that would only access the subset of the query languages available at all sources. In addition, the derived wrapper may adjust semantic differences of similar query operators.

The paper is organized as follows. In Section II we illustrate our approach with a motivating example that addresses the issue of integrating biological data sources such as GenBank and the use of a sequence similarity search tool such as BLAST in different contexts. Section III is devoted to the architecture of the proposed mediation system. Finally we conclude in Section IV.

II. MOTIVATING EXAMPLE

Biological data integration often is performed using hard-coding approaches. In this section we present a motivating example that illustrates how our architecture addresses the problem of data and applications integration. Issues raised by the integration scenario of this example are presented in [36].

A. Source schemas

We consider the following data sources: the Protein database [37], UniGene [38], PubMed [7], and GenBank [2]. We simplify the schemas of the data sources in order to illustrate our approach as follows.

PubMed is a service of the National Li-
library of Medicine that provides access to over
12 million MEDLINE citations and various
additional life science journals. PubMed pro-
vides multiple attributes, including:
- PubMed and MEDLINE identifiers.
- Citation information such as title, au-
thors, abstract, etc.
- Publication information such as: date,
  journal, page number, etc.
- Useful annotations such as MESH terms,
  GenBank or SWISS-PROT identifiers,
  etc.

The Protein database is available at NCBI
and contains sequence data from the trans-
lated coding regions from DNA sequences in
GenBank, EMBL and DDBJ as well as pro-
tein sequences submitted to Protein Infor-
mation Resource (PIR), SWISS-PROT, Pro-
tein Research Foundation (PRF), and Pro-
tein Data Bank (PDB) (sequences from solved
structures), and translations from annotated
coding regions in GenBank and RefSeq. Its
attributes include:
- Locus, definition, source, organism, se-
  quence, etc.
- References to papers in PubMed where
  the sequence was published, if any.

We denote by GenBank, the Nucleotide
database available at NCBI that contains se-
quence data from several sources including
GenBank, EMBL, DDBJ, RefSeq, and PDB.
The information provided by GenBank in-
cludes the following:
- An identifier describes a DNA sequence.
- References to papers in PubMed where
  the sequence was published, if any.
- The DNA sequence and the base count
  (number of A, C, G and T).

UniGene is a partition of GenBank se-
quences into a non-redundant set of gene-
oriented clusters. Each UniGene cluster con-
tains sequences that represent a unique gene,
as well as related information.
- The UniGene identifier that refers to
  the cluster.
- Identifiers corresponding to other data
  sources such as OMIM, LocusLink, Ho-
mologene, etc.
- Tissue type in which the gene has been
  expressed.
INTEGRATING BIOLOGICAL DATA AND TOOLS WITH BIS

- The Map location of the gene (chromosome, etc.)

In addition to rich and complex data sets, biological data providers often offer a variety of applications to better access and analyze the data they contain. PubMed offers Similar Articles, Domain Links, Genome Links, ProbeSet Links, Nucleotide Links, OMIM Linkstion approach typically performs the first query PopSet Links, Protein Links, SNP Links, and Structure Links. The protein database provides tools such as BLink, Related Sequences, Nucleotide, OMIM, PubMed, SNP, Taxonomy and Linkout. In addition NCBI provides access to a variety of BLAST programs.

B. Motivating queries

We motivate and illustrate our approach with the following two queries.

(Q1): Return all citations of PubMed published since 1995 that mention "heart" and refer to sequences of GenBank that are annotated as "calcium channel" [39]
(Q1) has two simple query execution plans. Both plans first accesses PubMed and retrieve citations published since 1995 that mention "heart". The first execution plan for (Q1) then extract all GenBank identifiers they contain, when the second plan uses the Nucleotide Link. Finally, both plans retrieve the information available in GenBank for each sequence and filter the ones that are annotated as "calcium channel". A traditional integratio-

(Q2): Return accession numbers and definitions of EST sequences that are sim-
lar (60% Identical over 50AA) to calcium channel sequences in the Protein database that have references published since 1995 and mention brain, [40].
A possible query execution plan for (Q2) first accesses PubMed and retrieve references published since 1995 that mention "brain". Then it extracts from all these references the Protein Identifiers and obtain the corresponding sequences from the Protein database whose function is calcium channel. Finally, it executes a BLAST call using a local wrapped
BLAST to retrieve similar sequences in the Protein database. An alternative plan for (Q2) consists in exploiting the BLAST program available at NCBI.

C. Schema Integration

BIS schema is an integrated schema of source schemas that exploits a detailed identification and description of relationships between the different data types and instances. BIS schema is expressed by Inter-schema Correspondence Assertions (ICA) that capture the correspondences between integrated sources. Scientific objects such as DNA sequences, clusters, genes and proteins are related to each other. For example, proteins are large molecules composed of one or more chains of amino acids in a specific order determined by the base sequence of nucleotides in the gene coding the protein. A gene is related to a protein by a one-to-one mapping scientifically characterized (from gene to protein) by the two successive steps: transcription of DNA into RNA and translation. Genes are DNA sequences whereas, on the other hand, not all sequences are genes. All these correspondences between scientific objects are expressed by ICAs thus provide life scientists to query integrated scientific data exploiting these relationships among scientific objects. Some of these relationships may be explicitly available at the integrated remote resources by the mean of an hyperlink or other computed index, whereas other relationships may not be available at the remote source while computable locally.

An example of an ICA rule is given with (C1) below:

\[ \text{UniGene.Cluster} \subseteq \text{GenBank.SET(identifier)} \]

ICA clauses reconcile representation discrepancies, therefore provide some "semantic glue" to provide users a transparent access to integrated resources. The ICA rule shown above enables the representation of the semantic correspondence between UniGene clusters and GenBank sequences and makes it available to the life scientists queries the two sources through BIS. Processing a query that involves
clauses C1 relies on the capability of the system to use schema correspondences expressed by means of ICA clauses. BIS uses the concept of derived wrapper introduced in Section III to evaluate ICA rules at query execution time.

D. Resources Integration

A system integrating biological data but failing to integrate important biological resources and or tools would be of little interest to life scientists. The approach presented in this paper offers a flexible way to integrating biological resources as illustrated with the following scenarii. For instance, biological data providers often supply various tools (E-Utilities at NCBI). Queries against BIS may invoke these tools as illustrated with (Q2). Suppose a life scientist wishes to "retrieve accession numbers of sequences that are annotated with "brain" and are similar (60% Identical over 50AA) to sequences in source X". There are two possible situations:

- Scenario 1: Source X provides a BLAST program.
- Scenario 2: Source X does not provide a BLAST or similar application.

Our approach can answer the query in these two situations. In scenario 1, the mediator knows that Source X provides a BLAST program and sends the BLAST call to resource X. In scenario 2, source X does not provide a BLAST program, the mediation retrieves the sequences annotated by calcium channel in source X and performs the BLAST locally against the virtual (cached) source composed of the sequences of Source X with a derived wrapper.

III. BIS Mediation System

A. Derived wrappers

We introduce the notion of derived wrapper that captures additional query capabilities either to reconcile schema discrepancies, to compensate capabilities lacking at a source, or to adjust an existing capability in order to make it homogeneous with other similar capabilities, wrapped at other sources. The use of derived wrappers extends traditional mediation approaches.
Standard wrappers access query capabilities provided at the integrated source as in mediation approaches. Derived wrappers are local applications that process cached data. Both types of wrappers are exploited similarly by the overall mediation as illustrated in Figure 1.

B. Overall architecture

Figure 2 describes the functional architecture of BIS developed at the Université de Provence, France, to seamlessly integrate both heterogeneous data and applications. The system is built on top a relational mediation system, namely LeSelect developed at INRIA [41]. LeSelect has a distributed peer-to-peer architecture; relational data sources are published on a LeSelect mediator by registering them as with a data wrapper connected to the mediator. A user query is formulated in SQL. Data source wrappers are built by the Wrapper Factory module in using a wrapper definition file (WDF). This file is either created manually by a publisher (data sources are available when the server is launched), or automatically by a program wrapper, in case we need access to new data sources or perform some utility tool.

The BIS mediator is composed of a query processor which consists of three components: an Analyzer, an Optimizer, and an Execution module. The BIS mediator is in charge of analyzing the query, including various transformations that involve ICA rules, performing some optimizations, and splitting the query into sub-queries, passing them to the right
wrapper for execution.

The Analyzer module uses the ICA rules to solve aggregation conflicts. The resolution consists in rewriting the query to eliminate the conflicts and to express it against the local source schemas. The Optimizer component exploits the information about source feature types and wrapped capabilities to select an efficient execution plan by dividing the query in several sub-queries. It processes the user’s query and forwards the generated sub-queries directly to the Execution module as illustrated in Figure 2. The Execution module executes the sub-queries and integrates the results returned by the different wrappers.

The Execution module processes the sub-queries as follows. Sub-queries that correspond to existing sources are directly sent to the appropriate wrappers. Sub-queries requiring an operator that is not available at any of the integrated sources is processed differently. As each operator is implemented by a Program Wrapper, the Execution module triggers a job request P1 to execute the operation. Program P1 is processed by the Program Manager component which, in turn, builds the associated Program Wrapper. A virtual data source is created and populated by results returned by program P1. A derived wrapper is also created in order to access the cached data as a new data source.

C. Query Processing

We demonstrate BIS query processing with the execution of (Q1) as follows. BIS accesses PubMed and GenBank as well as the E-Utilities tools provided by NCBI.

1. Extract references’ IDs via ESearch.
   
   This is performed by the following LeSe-lect program P1.

   job execute /cgiESearch
   parameter db='pubmed'
   parameter term='heart'
   parameter mindate='2002'

2. Cache the results returned by the program P1 in four tables, among them the one containing IDs, and denoted
TabID illustrated in Figure 3.

3. Extract references to GenBank entries following the Nucleotide Link with the program P2 given below.

```plaintext
job execute /NucleotideEFetch
parameter db='nucleotide'
parameter retmode='xml'
input i is select Id as id from
/temporaryWrappers/wrapper_3726586012684323153/IdList
```

4. Filter the retrieved GenBank entries to obtain the output as represented in Figure 4.

Fig. 3. Table TabID

Fig. 4. Table TabID

BIS query processing involves other programs that we can not illustrate here due to a lack of space. However, in our approach, all these programs are "wrapped", providing the life scientist with a transparent access to many remote and/or local data sources and resources.
IV. CONCLUSION

Accessing the large number of disparate genomic data sources and resources distributed over the Web is a problem of increasing importance to the bioinformatics community. BIS is a non-materialized data integration system which extends existing data integration systems to accommodate the needs of life scientists. The system complies with a standard mediation / wrapper architecture described in the literature, although most of the systems / prototypes do not deal with genomic data. To extend traditional mediation approaches to more flexibility for integrating applications, we introduce the new concept of derived wrapper which is used to integrate available tools such as NCBI E-Utilities. A derived wrapper is a wrapper of a virtual source that corresponds to a local data source (buffer) containing results of the execution of a local application. Derived wrappers capture different useful capabilities. First they support data integration by allowing the resolution of different data representation conflicts. In addition, they enable the representation and use of query capabilities that may not be provided by any or some of the integrated remote systems, while being available locally. Our approach is flexible, since a new source or new source capability may be registered easily with new wrappers (traditional and derived) and ICA rules.

The BIS system is developed and tested at the Université de Provence, France. Its future extensions will include: formal and effective data transformations, query optimization techniques to exploit information about integrated resources as proposed in [39], and the deployment of the software.

REFERENCES


[17] Z. Lacroix, “Scientific Data Integration: Wrapping Textual Documents with a Database View Mechanism and an XML Engine,” in IEEE In-
ternational Symposium on Bio-Informatics and Biomedical Engineering (BIBE), Washington, DC, November 2000.


