BioFast: Challenges in Exploring Linked Life Sciences Sources

Jens Bleiholder
Humboldt-Universität zu Berlin
bleiho@informatik.hu-berlin.de

Zoé Lacroix
Arizona State University
zoe.lacroix@asu.edu

Hyma Murthy
University of Maryland
hmurthy@umiacs.umd.edu

Felix Naumann
Humboldt-Universität zu Berlin
naumann@informatik.hu-berlin.de

Louiqa Raschid
University of Maryland
louiqa@umiacs.umd.edu

Maria-Esther Vidal
Universidad Simon Bolivar
mvidal@ldc.usb.ve

1 Introduction

An abundance of life sciences data sources contain data about scientific entities such as genes and sequences. Scientists are interested in exploring relationships between scientific objects, e.g., between genes and bibliographic citations. A scientist may choose the OMIM source, which contains information related to human genetic diseases, as a starting point for her exploration, and wish to eventually retrieve all related citations from the PUBMED source. Starting with a keyword search on a certain disease, she can explore all possible relationships between genes in OMIM and citations in PUBMED. This corresponds to the following query: "*Return all citations of* PUBMED *that are linked to an* OMIM *entry that is related to some disease or condition.*"

To answer such queries, biologists and query engines alike must traverse both the links and the paths (informally concatenations of links) through these sources, given some start object in OMIM. Figure 1 illustrates a subset of data sources at the National Center for Biotechnology Information (NCBI) that may be explored to answer the above query. All four data sources can be accessed at `http://www.ncbi.nlm.nih.gov`.

There are five paths (without loops or self-references among the sources) starting from OMIM and terminating in PUBMED. These paths are listed in Fig. 2. The large number of inter-related data sources, with dissimilar but overlapping content, and the large number of complex inter-relationships among these sources, raise a number of challenges in effectively and efficiently exploring life sciences sources.

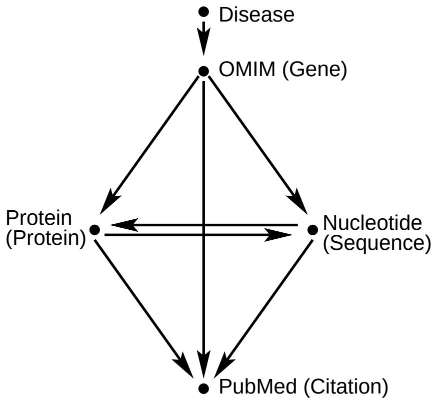

Figure 1: A source graph for NCBI data sources (and corresponding scientific entities)

In the BioFast project, we develop the infrastructure to support the biologist as she explores sources, links and paths, and develop methodology for effective data management and efficient query evaluation. Problems that are addressed include the following:

- **Query language for scientific exploration**:

(P1) Omim → PubMed

(P2) Omim → Nucleotide → PubMed

(P3) Omim → Protein → PubMed

(P4) Omim → Nucleotide → Protein → PubMed

(P5) Omim → Protein → Nucleotide → PubMed

Figure 2: Five paths from Omim to PubMed through the source graph of Figure 1

The challenge is the development of a high-level workflow-style language with appropriate operators and semantics that allow domain scientists to explore the contents and relationships captured in the sources. The operators and semantics of this language must be at a level of the biologist's procedures and experiments, which may then be translated into lower-level data manipulation operators. Evaluating the protocols expressed by the biologist will require both query optimization and query evaluation.

- **Semantics of links and paths**: In life science sources, links are implemented as physical references between data entries. To support more meaningful queries, these links must be enriched to capture semantics. Enrichment includes semantic labels and a more precise identification of the link's source and target elements within the data entry. Combined with the properties of links and paths, one can then perform a comparison of paths that is meaningful to the biologist.

- **Properties of links and paths**: The metrics (typically statistics) of links and paths may be used to characterize query results, e.g., to predict the cardinalities of query results along some path. These properties are useful to both biologists and data administrators as discussed later. The challenge is to identify and model interesting metrics and to efficiently measure them (e.g., sampling) or to correctly estimate them (e.g., statistical analysis). These properties can also be used to develop a cost model to determine the cost of evaluating a query plan.

- **Optimization for query answering**: Answers to explorative queries typically require traversing a multitude of paths among highly inter-linked sources. Each path differs in cost and benefit (result cardinality), making it non-trivial to choose the best path or set of paths. To compound the problem, the results of different paths overlap resulting in cost and benefit being considered not individually but for combinations of paths. The challenge is to evaluate queries efficiently (optimization) while at the same time producing answers that are meaningful to the biologist. Query planning and optimization of such queries poses many of the challenges that are addressed by database optimizers for mediation based architectures [4, 6]. Special challenges of life science queries are addressed in [1, 2, 3].

We present a simple model of life science data sources and then discuss our research in addressing the challenges of developing models for the properties and the semantics of links and paths.

2 Models for Life Science Data Sources

Life science sources may be modeled at two levels: the physical and logical level. The physical level corresponds to the actual data sources and the links that exist between them. An example of data sources and links is shown in Figure 1. The physical level is modeled by a directed *Source Graph*, where nodes represent data sources and edges represent a physical implementation of a link between two data sources. A data object in one data source may have a link to one or more data objects in another data source, e.g., a gene in GeneCards links to multiple citations in PubMed. An *Object Graph* represents the data objects of the sources and the object links between the objects. Each link in the source graph then corresponds to a collection of object links of the object graph, each going from a data object in one source to another object, in the same or a different source.

The logical level consists of classes (entity classes, concepts or ontology classes) that are implemented by one or more physical data sources or possibly parts of data sources. For example, the class *Citation* may be implemented by the data source PUBMED. Each source typically provides a unique identifier for the entities of a class and includes attribute values that characterize them. The following table provides a mapping from the logical classes to the physical data sources. A more detailed description of his model is in [5].

CLASS	DATA SOURCE
Sequence (s)	NCBI NUCLEOTIDE database EMBL Nucleotide Sequence database DDBJ
Protein (p)	NCBI Protein database Swiss-Prot
Citation (c)	NCBI PubMed

Table 1: A Possible Mapping from Logical Classes to Physical Data Sources

3 Properties of Links and Paths

As seen in Figures 1 and 2, a query on the four sources produces five potential paths that can be evaluated to produce results. It is important to characterize the properties of the links and thus obtain properties of the paths. These properties can be used for multiple purposes: One is for query planning and optimization, to determine the cost of evaluating the results of some path. Another is to estimate the size of the result, and the overlap among the different paths so that a user may choose to obtain answers from one or more paths, depending on the domain specific semantics of the paths.

A *Result Graph (RG)* represents the results to some specific query protocol, e.g., the bibliographical references (citations) from PUBMED that are linked to genes relevant to a given disease or medical condition. To create such RG, we fully explore all links and paths that exist between objects in the four sources, given some start set of objects in OMIM. We sample data by focussing on medical conditions: *cancer*, *aging*, and *diabetes*. A list of relevant keywords for each condition was used to retrieve relevant genes from OMIM. These genes constitute the starting set of objects. Each RG contains a collection of 140 to 150 OMIM records. Figure 3 shows the results of one such experiment for the condition *aging*, starting with 141 OMIM records along with the *measured* values for different paths through the result graph. Each edge label shows the link cardinality and each node label shows the number of distinct objects found by following those links (node cardinality). For example, the direct link from OMIMto PUBMEDhas 7149 link instances and reaches 7031 PUBMEDobjects, whereas the path through PROTEINhas link 6667 instances but only reaches 3275 PUBMEDobjects.

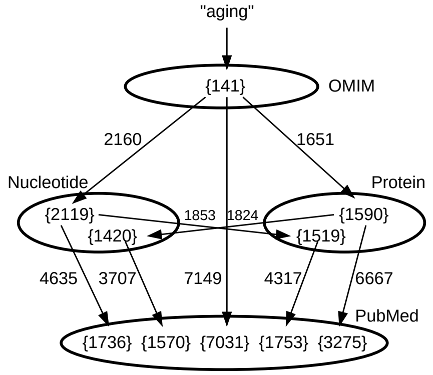

Figure 3: The result graph from an experiment on aging.

We describe a simple model to estimate the cardinality of an RG. For each link in the path, consider a start node (source) and target node (source) of a link, and properties such as the average outdegree of objects in the start source, the percentage participation of objects in the start source, the image cardinality of the target, etc. We develop a simple model to estimate the number of link instances and the the cardinality (the number of distinct objects of the target source) of a path. A simple model will make some assumptions such as uniform distribution of links across all objects, and independence of links. We estimate the result cardinality for a target source of a path as follows: Consider a path p through sources $S_1, \cdots, S_n$; let m_1 be the number of starting objects of source S_1. Assuming independence among object links and that no two outgoing links from a specific object reach the same target object (no overlap), we can calculate the number of objects reached for each source in p. To consider the overlap of object links, we must determine the likely number of distinct objects found in S_i, when randomly choosing m objects from all $C^{OG}(S_i)$ objects in S_i. Note that $C^{OG}(S_i)$ is the cardinality of source S_i, *or the number of data objects*. The probability to find exactly x distinct objects when picking m times from a set of $C^{OG}(S_i)$ in a source can be used to determine the expected number of distinct objects, i.e., the sum over all possible outcomes x multiplied by its probability. This is the estimated result cardinality. We can further enhance this model by noting that the image cardinality of source S_i is a more exact count of those objects in S_i that actually have an inlink along path p. Thus replacing $C^{OG}(S_i)$ with the image cardinality of S_i provides a more precise estimation.

The model described above assumed uniform distribution of the links across all objects. It also assumes object independence. For example, we assume that the probability that an object has an outlink, say from Protein to PubMed, is independent of the probability that it has an inlink, say from OMIM to Protein. However, in discussion with curators at NCBI, it is clear that such independence assumptions do not hold and that data objects and links typically violate such independence assumptions (Lash and Lipman 2003). For example, a curated object may be more likely to have both an inlink and an outlink. We refine our model to reflect the properties that we observe in these samled datasets. We define a path dependence factor (pdf) and a duplication factor (df) that reflect the properties of the sampled results graphs.
Path dependence factor (pdf): Let $p = S_i, S_{i+1}, S_{i+2}$ be a path of length 2. The *pdf* is the ratio of the number of objects that participate in links from S_{i+1} to S_{i+2} to the number of objects in the image of the link from S_i to S_{i+1}. *Duplication factor (df):* This is the ratio of the number of objects in $Si+1$ with an inlink from S_i to the link cardinality from S_i to S_{i+1}.

Using multiple training datasets (RGs), we can estimate the values for *pdf* and *df* for each of the links and paths. We note that it is often difficult to obtain such measures through random sampling in large graphs. We may utilize more sophisticated sampling techniques, e.g., comparing outdegree distributions for curated versus non-curated entries. Note that we may also determine that the values for *pdf* and *df* vary (are statistically significant) among different diseases and conditions, which would require additional sampling and tuning of the training data to obtain reasonably accurate estimated for *pdf* and *df*. We can extend the simple model (uniform distribution and independence), to recursively apply the measured values of *pdf* and *df*, along links and paths, to determine the number of objects reached on each path. Details of this model and experimental results are in [5].

To efficiently compute these metrics, we can define views corresponding to all paths of length greater than 1, e.g., all paths from OMIM to PubMed, or from OMIM to Protein, etc. and we can define queries to compute link and path cardinality. We must then address the problem of determining which views (paths) to materialize, and which views to use to compute each metric of interest. The problem is similar in nature to determining which views to materialize in a data warehouse environment, and optimization using views, given some query workload. In our example, the query workload corresponds to the paths of interest. Future work in the BioFast project concentrates both on the extension and generalization of the set of properties and on the usage of the presented properties for different scenarios, such as query optimization and data curation.

4 Enriching the Semantics of Links and Paths

As outlined before, data entries in different life sciences data sources have relationships that are expressed as links among them. However, these links are syntactically and semantically poor. The links are syntactically poor, because they exist only at a high granular level: the data entry. The links are semantically poor, because they carry no explicit meaning other than the fact that the data entries are somehow "related".

Links are added to data entries for many different reasons: Biologists insert them when discovering a certain relationship, data curators insert them to reflect structural relationships, algorithms insert them automatically when discovering similarities among two data items, etc. To represent such subtle and diverse relationships, simple links are not enough: Consider a Swiss-Prot entry with a link to an OMIM entry with a certain ID. In the flat structure of the Swiss-Prot entry this logical link is represented as a top-level attribute in the form of an OMIM ID, sometimes together with an HTML hyperlink to a data source storing that particular OMIM entry. A link in that form neither represents thepart of the Swiss-Prot entry the link refers to, nor the part of OMIM the link points to, nor does it represent the reason why the link was inserted. Biologists regarding the Swiss-Prot entry rely on their experience and can sometimes infer these link properties by closely and often time-consumingly examining both entries. Machines and algorithms cannot perform such analysis at the necessary level of detail and precision.

In the BioFast project, we are developing a general data model that allows the storage of links at a finer level of granularity, which allows users and machines to enrich the links. In the previous example, the link in question should not originate from the Swiss-Prot entry itself, but rather from a certain CC-DISEASE attribute within that entry. The link should also not represent a mere "relationship", it should rather be labeled as a "causal" relationship, telling humans and machines that the protein in question is known to cause the disease pointed to by the link.

A semantically enhanced link will comprise the following:

- A link identifier.
- A set of navigational paths to specify the origin of the link with respect to the parent data entry.
- A set of navigational paths to specify the target of the link with respect to the parent data entry containing this target.
- A link label or category and a link descriptor, possibly using some ontology.

Clearly, a data model alone is not enough, in particular because there already exist huge amounts of links that are not syntactically and semantically enriched. Hence, a next step is to explore methods to automatically or semi-automatically perform this enrichment by analyzing the linked data entries and by soliciting information from biologists in a user-friendly tool. For example, we may attempt to learn rules as follows: *A link from SwissProt to Nucleotide should go from the SwissProt element type A to the Nucleotide element type B given some particular annotation of the SwissProt entry.*

The benefits of this structural and semantic enrichment are numerous. Structural enrichment allows for a better and finer analysis of link structures as outlined in Section 3. It also spares biologists from having to infer or even guess the meaningful source and target of the link. Semantic enrichment is not only useful for humans reading data entries, but also allows semantically composing of multiple links to generate meaningful paths through life sciences data sources.

Next steps in the BioFast project include an inventory of link semantics that will list a set of possible semantic labels for links together with their respective domains for link-source and link-target. Next, we will explore techniques for automated and semi-automated link-enrichment, and finally investigate the composition of such semantics.

5 Conclusions

The issues that we present represents a starting point of understanding and exploiting Web life sciences sources and their relationships with one another. Each challenge represents a formidable task requiring further research. The success of this research will also require close cooperation with domain experts.

Our research has benefited from domain experts from IBM Life Sciences, NCBI and Humboldt-Universität.

Acknowledgements. This research is partially supported by NSF grants IIS0219909, EIA0130422 and IIS0222847, and the NIH National Institutes of Aging Grant 1 R03 AG21671-01. We thank Barbara Eckman of IBM Life Sciences and Stephan Heymann and Peter Rieger of Humboldt-Universität for insights into source properties, David Lipman and Alex Lash of NCBI for their expertise on NCBI data sources, Damayanti Gupta for data collection, and Marta Janer and Michael Jazwinski for identifying relevant keywords to construct the result graphs.

References

[1] B. Eckman, A. Kosky, and L. Laroco. Extending traditional query-based integration approaches for functional characterization of post-genomic data. *BioInformatics*, 17(2), 2000.

[2] B. Eckman, Z. Lacroix, and L. Raschid. Optimized seamless integration of biomolecular data. *Proceedings of the IEEE International Symposium on Bio-Informatics and Biomedical Engineering*, 2001.

[3] L. Haas, P. Kodali, J. Rice, P. Schwarz, and W. Swope. Integrating life sciences data - with a little garlic. *Proceedings of the IEEE International Symposium on Bio-Informatics and Biomedical Engineering*, 2000.

[4] Z. Ives et al. An adaptive query execution system for data integration. *Proceedings of the ACM SIGMOD Conference*, 1999.

[5] Z. Lacroix, H. Murthy, F. Naumann, and L. Raschid. Characterizing properties of paths in biological data sources. *To appear in the Proceedings of the DILS Conference*, 2004.

[6] V. Zadorozhny, L. Raschid, M.E. Vidal, T. Urhan, and L. Bright. Efficient Evaluation of Queries in a Mediator for WebSources. In *sigmod*, 2002.