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 Simón Bolívar
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[1].

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.

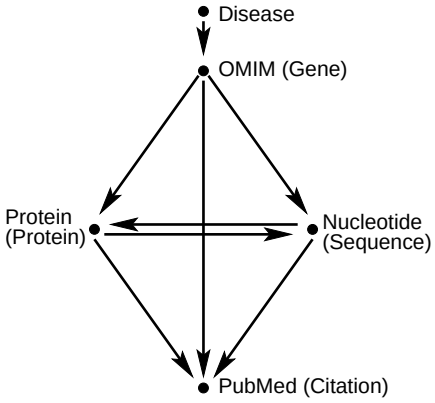

Figure 1: A source graph for NCBI data sources (and corresponding scientific entities)

- **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 biolo-

[1] All four data sources can be accessed at http://www.ncbi.nlm.nih.gov

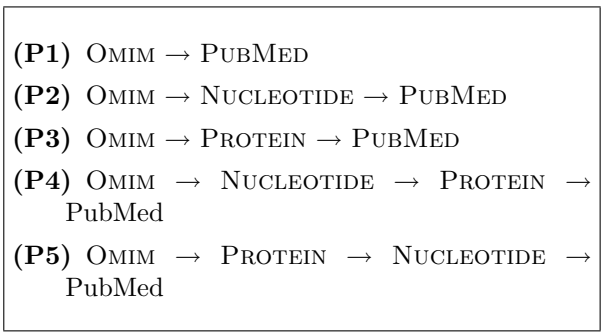

Figure 2: Five paths from Omim to PubMed

gists 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).

- **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, so cost and benefit must be considered not individually but for combinations of paths.

- **Semantics of links and paths**: In life sciences 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.

- **Query language for scientific exploration**: 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 the level of the biologist's procedures and experiments, which may then be translated into lower-level data manipulation operators.

In this extended abstract we present a simple model of life science data sources and then discuss our research in addressing these challenges.

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 this model is in [6].

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; see also Section 5).

We developed a simple model to estimate the properties of the paths. Given a start source and target source of a link, we used 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., to estimate path properties, such as the number of object link instances or object path instances (in the object graph), for some path, or the cardinality (the number of distinct objects of the target source) that are reached in a path. The simple model made some assumptions such as uniform distribution of links across all objects, and independence of links.

Based on sampled data from the four NCBI sources of Figure 1, for several diseases and conditions, we were able to determine that the prior assumptions do not hold. We then extended the simple model with metrics obtained from the sampled data. This included a *duplicate factor* for a link, i.e., the number of duplicate target objects given some set of object links, as well as a *path dependence factor*, i.e., given a path, the probability of an object participating in two consecutive links (inlink and and outlink) of the path. Our extended model based on the sampled statistics was able to better predict the properties of paths. Details of our model and experimental results are in [6].

Given the overlap of content of the sources and the connectedness of the sources, different paths will typically show some overlap. Determining the level of overlap is important both to a user as well as for a system that is evaluating the paths. We use our database of sampled data to study overlap among paths. Given the set of links of a source graph, we 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. We define queries to obtain statistics corresponding to these views, so as to determine path properties such as the cardinality of results (target objects) in each pair of paths, and the corresponding overlap between pairs of paths. We must now address the task of effective assessment of the overlap, including the problem of determining which views (paths) to materialize and which views to use to determine the overlap between any two paths. The problem is similar in nature to determining which views to materialize in a data warehouse environment, and query optimization using views, given some query workload. In our example, the query workload corresponds to the (pairs of) paths for which overlap is to be determined.

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 (see next section).

4 Query Answering and Optimization

We briefly ongoing research in the BioFast project related to query optimization and evaluation. A typical query posed by a biologist (or a query used to popu-

late an experiment or analysis pipeline) is quite complex, involving multiple entities distributed across many sources. Query evaluation may involve expensive predicates, e.g., similarity search, and other high-level operators such as *clustering*, *grouping*, *ranking*, etc. Thus, query planning and optimization of such queries poses many of the challenges that are addressed by database optimizers for mediation based architectures [4, 10]. Special challenges of life science queries are addressed in [1, 2, 3, 11].

In this discussion, we focus on the challenges of navigational queries, and on data overlap between the results of alternate paths. Consider a navigational query corresponding to a simple regular expression expressed over the logical scientific entities, such as those in Table 1. As discussed, a navigational query can be answered by multiple alternate paths in a source graph. The problem of determining if an edge in a graph occurs in a path that satisfies a regular expression has been shown to be NP hard [8]. We have developed an efficient algorithm based on a deterministic finite automaton to exhaustively enumerate all paths satisfying a regular expression [7]. It is polynomial for acyclic graphs and enumerates simple paths where an edge is not visited multiple times in a path. We further improve on this search by employing a breadth-first search strategy. The breadth-first search ranks sources based on a utility function that determines if this source will contribute towards the *benefit* of a path through this source. An example benefit criterion is the estimated number of distinct objects that are reached in the target source of a path (see previous section). Our research has shown that ranking sources based on their benefit contribution is not always a good heuristic to generate paths with the highest benefit and we are developing a heuristic that ranks all subpaths that have been generated.

In general, each path is associated with both a cost and a benefit and the objective is to generate some k best paths with the least cost and highest benefit. This is a multi-criteria optimization problem similar to the problem identified in [9]. A user may be interested in selecting those paths with the highest benefit to evaluate and obtain results. Alternately, a user may wish to choose multiple paths subject to an upper bound on evaluation cost or delay to obtain results. This requires the choice of some best k paths, i.e., the best *combination* of k plans. This task is compounded by the overlap of results of the target source along multiple paths, so that the benefit of a path is determined by the other paths chosen for evaluation. This problem corresponds to the budgeted maximum coverage problem [5], which has been studied in the logistics context of determining the optimal location of warehouses to cover multiple suppliers.

5 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 level. 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 insufficient. 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 with an embedded Omim ID, and possibly an HTML hyperlink to a data source storing that particular Omim entry. A link in that form neither represents the part of the Swiss-Prot entry that the link refers to, nor does it represent that part of the Omim entry to which the link points, nor does it represent the reason why the link was inserted. Biologists reviewing 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 model for links that allows the storage of links at a finer level of granularity and that allows users and machines to enrich the links. In the previous example, the link in question perhaps should not originate from the Swiss-Prot entry itself, but rather from the CC-DISEASE attribute within that entry. The link should also not represent a generic "relationship"; it should rather be labelled as a "causal" relationship, informing humans and machines that the protein may *cause the disease* described in the Omim entry.

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. The BioFast project will develop user-friendly tools to help a biologist to semi-automatically perform this enrichment, by analyzing the linked data entries and by soliciting information from biologists.

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 to semantically compose multiple links to generate meaningful paths through life sciences data sources. Finally, tools that are used to enrich links may also be used to help identify when the links are incorrectly inserted or are missing.

The BioFast project will develop an inventory of link semantics and will include 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, we will investigate the composition of such semantics.

6 Query Language for Scientific Exploration

An appropriate biological query language should enhance the scientist's querying ability by the following features:

- Provide an intermediate query language between scientific workflows and traditional query languages such as SQL.
- Provide high-level operators such as ranking and validation, which are currently not made directly available by traditional query languages and are often difficult to express (by complex queries).
- Constrain the evaluation of operators by various operator specific semantics.

An example high-level operator is the *Collection* operator; informally this operator has the function of collecting data from various data sources to increase the cardinality of explored entries, or to increase the characterization or the functional description related to entries, i.e., increase the cardinality of attributes. The corresponding semantics may involve maximizing (minimizing) the number of sources along a path, or maximizing (minimizing) the number of entries (objects in the target source) or the attribute cardinality either along the entire path or of the target source.

We note that understanding and generating the mapping from a desired experiment or data analysis protocol, expressed as complex workflows, to a set of operators and their semantics is often difficult and requires extensive domain expertise. The challenge is to define the operators and semantics at the appropriate level to be both able to express complex queries and to still be accessible to the biologist. The BioFast project will explore the definition of such high-level operators and their semantics, and will develop tools to allow domain experts to formulate queries using this language. The acceptance of the scientific query language and its semantics by biologists will depend on the enriched semantics of links and paths, as described in section refsec:semantics. Efficient evaluation of these queries will depend on research on the

metrics of links and paths (section 3), and on efficient query evaluation (section 4).

7 Conclusions

Research in the BioFast project is a starting point to understand and exploit life sciences sources and their relationships with one another. Only close cooperation with domain experts will ultimately lead to success of the Biofast project and the BioFast team is collaborating with domain experts from NCBI, Humboldt-Universität, and IBM Life Sciences, to address the challenges described in each of the sections of this paper.

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, Stephan Heymann and Peter Rieger of Humboldt-Universität and Terry Gaasterlanf 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 domain expertise on diabetes and aging research.

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] Barbara Eckman, Zoé Lacroix, and Louiqa Raschid. Optimized seamless integration of biomolecular data. *Proc. of the IEEE Int. Symp. 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. *Proc. of the IEEE Int. Symp. on Bio-Informatics and Biomedical Engineering*, 2000.

[4] Zachary G. Ives, Daniela Florescu, Marc Friedman, Alon Y. Levy, and Daniel S. Weld. An adaptive query execution system for data integration. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, pages 299–310, Philadephia, PA, 1999.

[5] Samir Khuller, Anna Moss, and Joseph Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.

[6] Zoé Lacroix, Hyma Murthy, Felix Naumann, and Louiqa Raschid. Links and paths through life sciences data sources. In *Proceedings of the International Workshop on Data Integration for the Life Sciences (DILS)*, Leipzig, Germany, 2004. to appear.

[7] Zoé Lacroix, Louiqa Raschid, and Maria-Esther Vidal. Efficient techniques to explore and rank paths in life science data sources. In *Proceedings of the International Workshop on Data Integration for the Life Sciences (DILS)*, Leipzig, Germany, 2004. to appear.

[8] Alberto O. Mendelzon and Peter T. Wood. Finding regular simple paths in graph databases. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, pages 185–193, Amsterdam, The Netherlands, 1989.

[9] Christos H. Papadimitriou and Mihalis Yannakakis. Multiobjective query optimization. In *Proceedings of the Symposium on Principles of Database Systems (PODS)*, Santa Barbara, CA, 2001.

[10] Vladimir Zadorozhny, Louiqa Raschid, Maria-Esther Vidal, Tolga Urhan, and Laura Bright. Efficient evaluation of queries in a mediator for websources. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, pages 85–96, Madison, WM, 2002.

[11] P. Mork, R. Shaker, A. Halevy, and P. Tarczy-Hornoch. Pql: A declarative query language over dynamic biological data. *Proceedings of the AMIA*, 2002.