Information Extraction and Integration from Heterogeneous Biological Data Sources

Cui Tao
Department of Computer Science
Brigham Young University, Provo, Utah 84602, U.S.A.
ctao@cs.byu.edu

Abstract

There are huge and growing amounts of biological data that reside in different online repositories. Most of these Web-based sources only focus on some specific areas or only allow limited types of user queries. To obtain needed information, biologists usually have to traverse different Web sources and combine their data manually. In this research, we propose a system that can help users to overcome these difficulties. Given a user's query within the the area of molecular biology, our system can automatically discover appropriate repositories, retrieve useful information from these repositories and integrate the retrieved information together.

1 Introduction

"Catalyzed" by world-wide research communities producing publicly available data, the volume of biological data is increasing at a rapid pace. To do activities such as perform background research for a specified research field, gain insights into relationships and interactions among different research discoveries, or build up research strategies inspired by other's hypothesis, biologists need a system that can help them discover, integrate, and extract the online information. The biological repositories online, however, are highly diverse in both granularity and variety. Different researchers focus on different levels of biological problems. This makes the online data sources focus on different granularities, such as, from genomic and protein sequences to protein activities, from cell structure to 2D or 3D structured data of huge molecules. Also, different sites present information various ways including images, plain text, tables, and information behind forms, and use different terminologies, different ID systems, or different units to describe the same concepts. Thus, automatic extraction and integration of online biological information is a challenging task.

We classify online biological data into two categories: *repositories* and *literature*. There are hundreds of available repositories of biological data, each with its own interface supporting different invocation, processing, and data semantics. [Bax, BGM, DBC], for example, all list hundreds of high-quality repositories of value to the biological community. These repositories have a large amount of information including, for example, DNA sequences, gene expressions, gene identification, intermolecular interactions, metabolic pathways and cellular regulation, mutation databases, protein sequences, RNA sequences, and large molecule structures. If a user needs to find detailed, particular information, the online literature is a better place to obtain the answer. Researchers publish a large volume of papers each year. These papers can either be found in an online literature database such as PubMed [pub] or by using online search engines from various sources or researchers' own Web pages. However, most of the documents are written in a human language rather than in a specific format

that computers would deal with more easily. These two sources are both critical and data-rich. Furthermore, sometimes the information a user needs exists across these two sources. A system that traverses only one source may not answer users' queries completely. Therefore, we propose a system to handle both of them.

Other solutions have been proposed that address some of these issues. [Won03] surveys recent technologies for integrating biological data. Systems such as EnsEmBL [Hub] and GenoMax [Gen] allow users to submit queries through their interfaces, but they only focus on a small range of biological concepts and problems and can only deal with a specific set of online repositories. It is simple and straightforward to add new data sources into systems such as SRS [EA96], but SRS does not allow users to write their own query. DiscoveryLink [Haa] has greater generality than SRS, EnsEmbl, and GenoMax and allows simple user queries, but it cannot deal with complex source data. Kleisli [DOTW97] has the ability to store and manage complex nested data, but it requires programming queries that are hard for biologists to write.

In our work, we propose a system that can automatically discover, extract, and integrate online biological data independent of the source and make it available for Semantic Web agents. Our system retrieves information based on our ontology extraction technology [ECJ+99, Emb04]. Section 2 introduces extraction ontologies and provides a short overview of how we generate an extraction ontology depending on user's query. Section 3 explains how our system can discover applicable online repositories, submit queries through form interfaces (if necessary), extract information of interest, and integrate it. Section 3 also discusses how we can use this same system to extract useful information from published papers. Section 4 concludes by summarizing the system's features and the critical work to be done to make the system successful.

2 Extraction Ontology

Our system works based on information-extraction ontologies. An extraction ontology is a conceptual-model instance that serves as a wrapper for a narrow domain of interest. The conceptual-model instance includes objects, relationships, constraints over these objects and relationships, descriptions of strings for lexical objects, and keywords denoting the presence of objects and relationships among objects. When we apply an extraction ontology to a Web page, the ontology identifies the objects and relationships and associates them with named object sets and relationship sets in the ontology's conceptual-model instance and thus wraps the recognized strings on a page and makes them "understandable" in terms of the schema implicitly specified in the conceptual-model instance. In our previous research, we have experienced with many ontologies for real-world applications such as cars and obituaries ([Emb04] summarizes our previous research). These experiments indicate that ontological conceptualization over recognized data items is a promising way to achieve the goal of semantic agreement over heterogeneous sources.

Our gene extraction ontology (GEO) contains various concepts (object sets) in the molecular biology domain in different granularities such as *Gene, Gene Name, Gene Locus Tag, Gene Sequence, DNA, DNA Sequence, RNA, RNA Sequence, Protein, Protein Name, Protein ID, Protein Activity, Protein Function Description, Mutant, Mutant Name, Mutant Indicator, Mutant Function, Source Organism or Species.* We define relationship sets between these concepts, i.e., *Protein has Protein Activity* and *DNA translates to RNA*. There are also aggregations or generalizations/specializations between object sets. For example, *Gene* is an aggregate of *Gene Name, Gene Sequence*, and *Gene Locus Tag*, and *Protein Activity* can be specialized to *Enzyme*, *Binding*, and many more such specializations. For each object set, there is a data frame that defines how the extraction

ontology recognizing target terms or phrases in source documents. A data frame contains either regular expressions or vocabulary terms or phrases that can help our system to recognize values for a concept. In order to build successful extraction ontologies for biological data, we need a trusted knowledge base that we can use as a thesauri. The Gene Ontology (GO) [GO] is a generally acknowledged tool that provides a controlled vocabulary for the description of cellular components, molecular functions, and biological processes. [1] Therefore, we use GO as our thesaurus of data-frame terms and phrases.

GEO is general and contains many concepts. A user query, however, is usually just about a few of these concepts. If we use the whole extraction ontology for the extraction and integration task, not only is it expensive in both time and space, but we are likely to retrieve much unwanted information and lose our focus on the items of interest to a specific user query. Therefore, we use a smaller, query-oriented extraction ontology (QEO) for each user's query. When a user submits a query, we decide which object sets and relationship sets we should include in the extraction ontology depending on the query as follows.

1. *Match query concepts with ontology concepts.* We first match the concept(s) in the query with object sets in our extraction ontology. Since a user may not use the same term we use to define a concept, our name matcher needs to match the corresponding concepts semantically. Since a user may also describe concepts at a different granularity than we do, the matching may be one to one, one to many, many to one, or even many to many.

2. *Augment recognized concepts to form a meaningful query-oriented extraction ontology (QEO).* We include all the object sets that match with concepts in the query. Augmentations include the following. If a matched object set is in an aggregation, we also add other components in the aggregation to the new ontology to facilitate Web querying. If the query references a relationship, but does not include all the objects in this relationship, we also add all the object sets in the extraction ontology. Any object set in a one-to-one correspondence with included concepts is also included. An augmented QEO makes our search better.

Consider the following query as an example: "Find *Aspergillus flavus*'s gene: *aflR*'s sequence, expressed protein's function and any mutant that inhibits this gene." We first need to match the concepts in this query with object sets in GEO. We obtain seven matches: "*Aspergillus flavus*" $\leftrightarrow$ *Source Organism or Species*, "gene" $\leftrightarrow$ *Gene*, "aflR "$\leftrightarrow$ *Gene Name*, "sequence" $\leftrightarrow$ *Gene Sequence*, "protein's function" $\leftrightarrow$ *Protein Function Description*, "mutant" $\leftrightarrow$ *Mutant*, and "inhibits" $\leftrightarrow$ *Mutant Function*. We augment these recognized concepts to form our QEO as follows. Along with *Gene name* we add *Gene Locus Tag* and *Gene Sequence* as aggregate components for *Gene* and *Protein Name* and *Protein ID* as aggregate components for *Protein*. Since *Mutant Indicator* is in the relation of *Mutant Indicator* indicates *Mutant* of *Source Organism or Species* that has *Mutant Function* on *Gene*, we include all of these object sets. After generating an extraction ontology, the system is ready to "query" the online resources and retrieve the useful information.

3 Query Oriented Extraction and Integration

Given a user's query, our system can automatically discover appropriate repositories, retrieve useful information from these repositories, and integrate the retrieved information together.

[1] Although called an ontology, the Gene Ontology is not an extraction ontology. It is a controlled vocabulary. GO contains 1364 components, 7268 functions and 8026 process terms as of December 1, 2003.

Our system contains following three components.

- *Source Discovery.* There are large amounts of online resources. How does the system find appropriate repositories and papers that contain the answer to a query? There are several classical tools for document classification and source discovery such as the Vector Space Model (VSM) [RW86] and Bayesian networks [FGG97]. Our system discovers appropriate resources by using an approach that combines classical document classification methods with ontology extraction techniques. VSM requires a set of index terms that represents the domain of interest. It calculates similarities between index terms and a test document to determine if the document belongs to the domain of interest. One big problem with this approach is to determine good index terms that represent a user's query and how to group index terms depending on semantic equivalence. Our extraction ontology provides a good solution to this problem. Since a user's query is represented in the context of a QEO, we can use all the terms defined by the data frames for the query in this ontology as index terms. Terms that describe the same concept are in the data frame under one object set. Therefore, they are already clustered .

  Another problem with VSM is that it assumes independence among all the index terms. But there are relationships between index terms and these relationships sometimes affect the classification, especially when there are many concepts mentioned in the query. In this case, we use a Bayesian net (BN) classifier. A BN classifier, however has its own practical problem: it is usually hard to obtain the structure of the net. Our ontology also provide a good solution to this problem. The ontology itself defines the relationships between different concepts. The graphic version of a QEO is a network. Therefore, we can use this "ontology net" as the structure of the Bayesian net.

- *Retrieval of Information Hidden Behind Forms.* Some of the online repositories allow users to submit queries through their interface. This helps users find the information of interest easier and faster. Different repositories, however, usually provide different interfaces that require users to submit their queries in different Web specific ways. It is tedious for users to visit dozens of sites for the same application and fill out different forms provided by each site. Enabling automated agents and Web crawlers to interact with form-based interfaces designed primarily for humans would be of great value. [Che] proposes an approach that can achieve this goal. Given a user's query, the system analyzes each online responsory's form, matches field names in the form with object sets in the extraction ontology using our ontology based schema matching techniques [Xu], then fill out the form depending on a user's query, and finally submits the form to obtain information the user needs.

- *Data Extraction and Semantic Schema Matching.* Our ontology-based information extraction technology can extract information from unstructured, semi-structured, and structured data. [ECJ+99] explains how to extract data of interest from unstructured or semi-structured source files. [ETL02] discusses an approach that can integrate information from heterogeneous tables. In this approach, we first pair and adjust attribute-value pairs in the source table. We then perform extraction from the formed attribute-value pairs to the target. Given the recognized extraction (which need not be 100%), the system can infer general mappings from source to target and then extract more information. We tested these approached in different application domains such as car ads and cell phone sales. We believe that the same techniques also apply to the biological domain.

Gene	Gene Name	Gene Locus Tag	Gene Sequence	Protein
001	alfR	*AN*7820.2	Aspergillus nidulans contig 1.132 226012-227313	001

Protein	Protein Name	Protein ID	Protein Function Description
001	AFLR_EMENI protein Sterigmatocystin biosynthesis regulatory	1ajy	zinc ion binding

Mutant Indicator	Mutant	Source Organism or Species	Mutant Function	Gene
tan, afl-4, pdx6	strain 241	a. flavus	block	001
wA, methG1, biA1; stcE::argB	TSS40	a. nidulans	defective	001

Figure 1: Sample Result of the Example Query

As an example to illustrate how these three components cooperate together, we further consider our query: "Find *Aspergillus flavus*'s gene: *aflR*'s sequence, expressed protein's function and any mutant that inhibits this gene." Given the QEO for the query obtained in Section 2, we use the source discovery component to find possible repositories that answer the query. We then go to these online repositories, use our ability to retrieve information hidden behind forms. In this initial stage, the only thing we know from the query is the *Gene Name* ("aflR"), *Mutant Function* ("inhibits"), and *Source Organism or Species* (*Aspergillus flavus*). For repositories such as [ANA] that allow searches by multiple concepts including gene name, we can automatically fill out forms and retrieve information. What we retrieve from [ANA] are *Gene Locus*, *Gene Sequence*, and *Protein Name*. Now we know more information about what we are looking for, and we can automatically fill out more forms to retrieve more information. After we try a certain number of forms, we may still not be able to answer all the questions implied by the query. In our example, we cannot find answers for "any mutant that inhibits this gene." In this case, we search for online literature because it usually contains detailed and specific information that online repositories do not contain. We use the same technique to find useful papers and extract needed information from them. [AFG], for example, is a source we can use to answer the query. Figure 1 shows the sample result the system can obtain.

4 Summary

We are proposing a system that can automatically discover appropriate repositories, retrieve useful information from these repositories and integrate the retrieved information. Our system depends on a dynamically constructed query-oriented extraction ontology. Our System has three main components: source discovery, retrieval of information hidden behind forms, and data extraction and integration. We have already developed some techniques for each of the components. In the future, we will augment these techniques and do experiments in the domain of biology.

References

[AFG] Abstracts from the 19th fungal genetics conference. http://www.fgsc.net/asilomar/secmetab.html.

[ANA] Aspergillus nidulans autocalled gene search. http://www.broad.mit.edu/annotation/fungi/aspergillus/geneindex.html.

[Bax] A. D. Baxevanis. The molecular biology database collection. http://www3.oup.co.uk/nar/database/.

[BGM] Biology/genetics/microbiology databases. http://www.edae.gr/bio-databases.html.

[Che] X. Chen. Query rewriting fo ext acting data behind html forms. In *Technical Report, Brigham Young University, Provo, Utah, 2002.*

[DBC] The public catalog of databases. http://www.infobiogen.fr/services/dbcat/.

[DOTW97] S. B. Davidson, G. C. Overton, V. Tannen, and L. Wong. Biokleisli: A digital library for biomedical researchers. *International Journel on Digital Libraries*, 1(1):36–53, 1997.

[EA96] T. Etzold and P. Argos. Srs: Information retrieval system for molecular biology data banks. 266:114128, 1996.

[ECJ+99] D.W. Embley, D.M. Campbell, Y.S. Jiang, S.W. Liddle, D.W. Lonsdale, Y.-K. Ng, and R.D. Smith. Conceptual-model-based data extraction from multiple-record Web pages. *Data & Knowledge Engineering*, 31(3):227–251, November 1999.

[Emb04] D.W. Embley. Toward semantic understanding – an approach based on information extraction. In *Proceedings of the the Fifteenth Australasian Database Conference*, Dunedin, New Zealand, January 2004. to appear.

[ETL02] D.W. Embley, C. Tao, and S.W. Liddle. Automatically extracting ontologically specified data from HTML tables with unknown structure. In *Proceedings of the 21st International Conference on Conceptual Modeling (ER'02)*, pages 322–327, Tampere, Finland, October 2002.

[FGG97] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163, 1997.

[Gen] GenoMax. http://www.informaxinc.com/solutions/genomax.

[GO] Gene ontology (go) consortium. http://www.geneontology.org/.

[Haa] L.M. Haas. Discoverylink: A system for integrated access to life sciences data sources. *IBM Systems Journal*, 40(2).

[Hub] T. Hubbard. The ensembl genome database project. *Nucleic Acids Research*, 30(1).

[pub] Pubmed by nlm. http://www.ncbi.nih.gov/entrez/query.fcgi.

[RW86] V. V. Raghavan and S. K. M. Wong. A critical analysis of vector space model for information retrieval. *Journal of the American Society for Information Science*, 37(5):279–287, 1986.

[Won03] L. Wong. Technologies for integrating biological data. *Briefings in Bioinformatics*, 3(4):389–404, 2003.

[Xu] L. Xu. Source discovery and schema mapping for data integration. In *Technical Report, Brigham Young University, Provo, Utah, 2003.*