Biological Data Extraction and Integration — A Research Area Background Study

Cui Tao

May 24, 2005

Abstract

My research field is highly diverse. It interweaves many different areas in information technology and bioinformatics. The system I propose to implement can automatically locate, understand, and extract online biological data independent of the source and also make it available for Semantic web agents. This research field requires background knowledge from (1) Information Extraction, (2) Schema Matching, (3) the Semantic web, (4) Data Integration, and (5) Bioinformatics.

1 Information Extraction

Currently, with the fast development of the internet, both the amount of useful data and the number of web sites are growing rapidly. The web is becoming an increasingly useful information tool for computer users. However, there are so many web pages that no human being can traverse all of them to obtain the information needed. Even in the narrow domain of molecular biological data, no human can traverse all the pages that may be of interest for finding needed information. A system that can allow users to query web pages like a database is becoming increasingly desirable. One possible strategy is to extract useful information from different web pages to populate databases for further handling. I survey information extraction techniques in the following three categories: (1) traditional information extraction; (2) hidden web crawling; and (3) biological data extraction.

- *Traditional information extraction.*

  For traditional information extraction, I present five major data extraction tools. Each tool represents a different major way of doing information extraction.

  Lixto [BFG01] is a tool for supervised wrapper generation and automated web information extraction. It generates wrappers semi-automatically and interactively by creating patterns in a hierarchical order. The user can define extraction patterns

through the interface and further refine them until satisfied with the elements identified by the system. Then Lixto uses the wrapper to extract the relevant information from an HTML document and translate it into XML which can be easily queried and further processed. Lixto has a friendly interface and does not require users to know any specific language. However, it is not robust to changes in web pages and does not work well with unstructured data.

ROADRUNNER [CMM01] does fully automatic wrapper generation. It does not need any interaction with the user during the wrapper generation process. ROADRUNNER compares two HTML pages from one web site and analyzes the similarities and dissimilarities between them in order to discover the pattern of how this web site presents data. The system discovers data fields by string mismatches and discovers iterators and optionals by tag mismatches. For more complex cases, the system may need more than two pages to capture more accurate structural variations. Although this approach is fully automatic, it does not generate robust wrappers and thus has to generate one wrapper for each web site. Another problem is it only works for web pages that are highly regular, usually only those that are generated automatically.

SRV [Fre98] and RAPIER [CM99] both combine NLP techniques with machine learning algorithms. SRV is a general-purpose top-down learner for information extraction. SRV learns extraction rules and extracts useful information from text documents based on a set of token-oriented features. There are two basic varieties of token-oriented features: simple and relational. A simple feature is a function mapping a token to some discrete value, such as length, character type, orthography, part of speech, or lexical meaning. A relational feature considers relationships between tokens, such as adjacency or linguistic syntax. SRV is not robust to changes; its training documents need to be labelled; and it does not work well as a "multiple slot" filler. [1] RAPIER is a bottom-up relational learner of pattern-matching rules for information extraction. The pattern-matching rules are indexed by template name and slot name. Each rule consists of three parts: pre-filler pattern, slot filler pattern, and post-filler pattern. A slot filler pattern matches the information that needs to be extracted, and a pre-filler and post-filler match the context of the information of interest. An extraction pattern considers features such as word lengths, symbols, part-of-speech tags, and semantic classes. RAPIER induces the extraction pattern from a pre-tagged training set. It is a "single-slot" approach and can only work with free text.

[1] For some information extraction tasks, an attribute may have zero (missing) or multiple instantiations in a record. A wrapper that can extract one tuple of interest is called a single slot filler and a wrapper that can extract a list of tuples is called a multiple slot filler.

BYU Ontos [ECJ+99] is an ontology-based data extraction system. A domain specific extraction ontology describes the data of interest by using objects, relationships, and data frames which contain data-value recognizers. The ontology guides the extraction process by providing conceptual expectations which can be matched using pre-specified heuristics. This approach is robust to changes in source pages and can extract and integrate information from different web sites in the same application domain. It works for unstructured, semi-structured, or structured source documents that require "multiple slot" filling. The drawback of this system is that it requires human experts to build extraction ontologies manually.

- *Hidden web crawling.*

  Traditional information extraction tools only work on the publicly indexable web (web pages reachable purely by following hypertext links). However, large numbers of web pages are hidden behind search forms. These pages are dynamically generated through searchable online databases according to users' queries submitted through the search forms. HiWE (Hidden web Exposer) [RGM01] is a hidden web crawler that can crawl the hidden web according to a user's query. When it encounters a form page, the crawler first builds an internal representation of the form. It then tries to match the internal form representation with the concepts in a task-specific database. Once concepts are matched, HiWE can assign values to each internal form field according to the database. HiWE uses value assignments to fill out and submit the search form. It then can retrieve the information hidden behind the form.

- *Biological data extraction.*

  Current biological information extraction approaches mainly extract data from plain text such as online abstracts and articles. Systems such as [KRMF00] recognize biological terms such as protein and gene names. Other systems such as [GDAW03] focuses on relationships between biological terms/elements, such as interactions between proteins and amino acid residues in protein molecules.

  Recognizing biological terms from a plain text document is a non-trivial problem. It is, however, one of the first steps toward achieving the goal of biological information extraction. The approaches to named-entity extraction can be divided into two categories: rule-based and dictionary-based. Rule-based approaches generate heuristic rules based on text features such as morphologic characteristics, part-of-speech tagging, or keywords. Dictionary-based approaches consist of first constructing named-entity dictionaries and then detecting dictionary terms in documents. Rule-based approaches are particularly useful in identifying new names. However, if a biological

object has multiple synonyms, rule-based approaches are not able to unify them. This problem can be solved by the dictionary-based approaches. Here I introduce a system for protein and gene name recognition which is mainly a dictionary-based, but also considers spelling variations in names to recognize biological terms [KRMF00]. The system works based on BLAST (Basic Local Alignment Search Tool) which provides a method for rapid DNA and protein sequence comparison and a database for gene and protein names. First an exhaustive list of gene and protein names is translated into an alphabet of DNA sequences by substituting each character in the name with a predetermined unique nucleotide combination and then the encoded names are imported into BLAST. Once the system has a source article, the system encodes it using the same nucleotide combination. The system then matches the translated article against the nucleotide representation of gene and protein names. BLAST finds any exact match; it also considers similar sequences. Therefore, this tool can find both exact names and names that are closely similar to the names in the dictionary.

In addition to extracting biological element names, it is also important to extract relationships among these biological elements. PASTA (Protein Active Site Template Acquisition) [GDAW03], for example, is one of the tools to automatically extract amino acid residues in protein molecules from online articles and abstracts. A PASTA template stores information about an entity, a relation, and an event. The system fills out the slots in a template using the following four steps. (1) In a text preprocessing step, the system analyzes each section in a source document and discards those sections that are not related to the domain of interest. It also splits those sections that are related to the domain of interest into sentences and character sequence units. (2) In a terminological processing step, the system identifies and classifies instances of the term classes by analyzing the morphological features of each term and looking them up in biological databases. It also combines related adjacent terms into phrases. (3) In a syntactic and semantic processing step, the system builds a "semantic" representation of the text on a sentence-by-sentence basis by using NLP syntactical and grammatical analysis. (4) In a discourse processing and template extraction step, the system fills out the templates, links information from sentences, and merges the related information together.

2 Schema Matching

Automatic schema matching is an important problem for many database applications such as data integration, interoperability resolution, and ontology alignment. For data integra-

tion, data extraction, and data source understanding over heterogenous biological data, schema matching plays a central role. It is also useful for ontology evolution. Previous schema matching methods can be classified as individual matchers vs. combined matchers, schema-based matchers vs. instance-based matchers, learning-based matchers vs. rule-based matchers, and element-level matchers vs. structure-level matchers [RB01]. Many schema matching approaches combine several methods together. Here I introduce four approaches that cover most of these methods.

The LSD (Learning Source Description) system is a semi-automatic learning-based approach [DDH01]. After a small set of data sources have been manually mapped to the mediated schema, LSD uses these mappings together with the sources to train a set of learners. Then LSD finds semantic mappings for a new data source by applying the learners. This system learns from both schema-level and instance-level information. LSD consists of four major components: a base learner, a meta-learner, a prediction converter, and a constraint handler. It operates in two phases: training and matching. In the training phase. the system trains base learners on manually created training examples. Different base learners require different sets of training examples. In the matching phase, the system uses the trained learners to match new source schemas. The meta-learner and the prediction converter combine the results of each base learner, and then the constraint handler takes the overall predictions and outputs 1-1 mappings (both element-level and structure-level). The authors of LSD recently developed another learning-based approach called GLUE [DMD$^+$03]. GLUE tries to match concepts in different ontologies based on well-founded notions of semantic similarity, expressed in terms of joint probability distributions on the concepts involved. GLUE only supports 1-1 mappings by selecting the candidate with the highest similarity for each concept. CGLUE is an extended version for GLUE that can work on complicated mappings.The matching accuracy of CGLUE, however, is only about 50%.

Cupid is a rule-based matcher that does both element-level and structure-level matching [MBR01]. This system first models the interconnected elements of a schema as a tree. Then it calculates similarity coefficients between attributes of the two schemas and deduces a mapping from those coefficients. The coefficients are computed in two steps: linguistic matching and structural matching. The linguistic matching step considers linguistic features of the elements, such as name, data type, and domain, and computes a linguistic similarity coefficient, *lsim*, between each pair of elements. The structural matching step matches schema elements based on the similarity of their contexts and vicinities. This step depends in part on the linguistic matching step. The result of this step is a structural similarity coefficient, *ssim*, between each pair of elements. Finally, the system calculates a weighted

similarity, *wsim*, by a weighted average of *lsim* and *ssim* and creates a mapping by choosing pairs of schema elements whose *wsim* is maximal.

Different schema matchers cover different problems and situations and have their own advantages and disadvantages. COMA (COmbiing MAtch) has an extensive library of matching algorithms and supports different ways for combining the results [DR02]. New matchers are easily added to the library. COMA is also an evaluation platform that can compare the effectiveness of different matchers in the library. COMA has three phases: an optional user feedback phase, a phase for execution of different individual matchers, and a phase for the combining the different match results. In the user feedback phase, the user can interact with the system to specify the match strategy, define match or mismatch, and accept or reject the proposed matching candidates. In the individual match phase, COMA executes different matchers chosen from the matcher library. For the combination phase, COMA combines matched results from the individual matchers by aggregating matcher-specific results and then selecting from among the match candidates depending on the similarity values.

3 The Semantic web

The semantic web is a mesh of information linked in such a way as to be easily processable by machines. It is an efficient way of representing data on the web, or as a globally linked database [BLHL01]. The semantic web is an emerging concept. It is different from the current web, which is only designed for humans to read, because the semantic web is also for computer programs to manipulate meaningfully. The semantic web extends the current web and will allow computers to "understand" the semantic meaning of web content, thus better enabling computers and people to work in cooperation. If we can transfer the biological online resources to the semantic web, it will be much easier for us to obtain more valuable information across heterogenous sources. With the semantic web, we can obtain intelligent information services, personalized web resources, and semantically empowered search-engines over biological data.

Semantic web content is "data + metadata" [DK03]. Data can be structured data, semi-structured data, or unstructured data. Metadata is data that describes data. The semantic web enables interoperability at the semantic level. Semantic interoperability requires standards not only for the syntactic form of documents, but also for the semantic content. Proposals aiming at semantic interoperability are the results of recent W3C standardization efforts: XML, RDF, and most recent one, OWL. XML is designed to improve the functionality of the web by providing more flexible and adaptable information identifi-

cation. It allows user to define their own tags. XML is aiming at the structure of documents and does not impose any common interpretation of the data contained in the documents. RDF is a framework for describing and interchanging metadata, as well as describing data about web resources. The basic construct in RDF is an object-attribute-value triple. All objects are independent entities. Semantic units are given naturally through its object-attribute structure. A domain model, defining objects and relationships for a domain of interest, can be represented naturally in RDF. OWL extends RDF/XML exchange syntax and an abstract frame-like syntax, and adds Description Logic style model theory to formalise the meaning of the language [HPSH03]. Thus, OWL can provide more expressive descriptions and more precise semantics.

When the idea of the semantic web has been realized, much of the "intelligence" that can now only be done by humans can be provided automatically by computers. One way to transfer a current web document to a semantic web document is through semantic annotation. Many researchers are working on semantic annotation of documents with respect of ontology and entity knowledge base. [KPT+04], for example, presents a tool called KIM (Knowledge and Information Management) that automatically annotate unstructured and semi-structured content, mostly name entities.

4 Data Integration

Data Integration is a broad topic. There are many approaches that focus on how to integrate information from heterogonous sources. Here I only focus on several data integration systems in the biological domain.

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 performing background research for a research field, gaining insights into relationships and interactions among different research discoveries, or building up research strategies inspired by other's hypotheses, biologists need a system that can integrate online bio-information. [HK04] summarizes the challenges of integrating biological sources. Online biological repositories are highly diverse in both granularity and variety. Different researchers focus on different levels of biological problems. Thus online data sources focus on different granularities and use different terminologies, different ID systems, or different units to describe the same concepts. Most of these sources are unstable and unpredictable. Independent developers modify their designs and schemas, remove or add data dynamically, or occasionally block access to their sources for maintenance or other purposes. Some of the sources provide interfaces through which a user can query a source. But each individual source provides

different interfaces and only allows certain types of queries to be asked. An integration system needs to resolve all these problems. Automatic integration of online biological information is thus a challenging task. Here I provide an overview of several systems. These systems are chosen to span from specialized solutions to increasingly general solutions.

The Sequence Retrieval System (SRS) [EA96] is a keyword based retrieval system. SRS provides a graphical interface across a broad range of biology resources, including biological sequences, metabolic pathways, and literature abstracts. When a biologist submits some keywords and constraints, the system retrieves the relevant documents for the user. The returned results are a simple aggregation of records that matched the query. Therefore, SRS is closer to a keyword-based retrieval system than an integration system. In addition, SRS has strict requirements for source documents, and it only works with relational databases.

BioKleisli [DOTW97] is one of the earliest information integration system over biological data. The system queries and combines information from heterogeneous data sources and application programs. It is a mediator system encompassing a nested relational data model, a high-level query language, CPL (Collection Programming Language), and a query optimizer. BioKleisli does not use any global schema or ontology over which a user can formulate queries. A query attribute is bound to a matched attribute in a single source, so there is no integration across different sources. Furthermore, the query optimization only focuses on reordering the Boolean operations. No optimization based on source content is performed.

DiscoveryLink [Haa01] is a system that provides users with virtual database access to different sources. DiscoverLink has two key components: a wrapper architecture and a query optimizer. The wrapper architecture maps the query fragments submitted to the wrappers into source queries that can be processed by each data source and retrieves the result returned by each source. The query optimizer examines a query bottom-up. It considers the speed of various sources, their network connections, and the size of their data to predict the costs of different plans. DiscoveryLink, however, cannot deal with complex source data such as nested data. Most biological data, unfortunately, is highly nested. Therefore, there is a significant amount of mismatch between most data sources and DiscoveryLink. Furthermore, it is hard to add new data sources or analysis tools to DiscoveryLink. In addition, DiscoveryLink requires SQL as its query language, which is not easy for biologists to write. Another drawback of DiscoveryLink is that it uses only C++, which is not an ideal language for a web wrapper.

TAMBIS (Transparent Access to Multiple Bioinformatics Information Sources) [SBB+00] is a retrieval-based information integration system for biologists. TAMBIS works based on TaO (the TAMBIS global domain Ontology), a domain ontology for molecular biology

and bioinformatics represented by a description logic language. The system uses TaO to describe a visual interface and a global schema against which a user can ask intersource queries. TAMBIS has three layers: the conceptual model, the mapping model, and the physical model. The global ontology is a unified conceptual-level representation of its registered component resources. It provides a global schema as well as an abstract framework for relating, reconciling, and coordinating the concepts in the sources. Based on the conceptual model, a source independent query can be formulated. In addition, some of the queries can be answered intensionally based on the ontology alone. The mapping model converts a query phrased in terms of the conceptual layer into executable plans in terms of each source. Currently, this step is executed manually. In addition, TAMBIS only considers five source repositories. The physical model submits executable plans to different sources and retrieve the results. Unlike DiscoveryLink, TAMBIS offers a global schema and data reconciliation. It also hides the sources from the users so that it is more "transparent." Although TAMBIS is more of an upper level solution than DiscoveryLink, its mapping model is implemented manually. Therefore it is not robust to changes in a source. In addition, its interface is complicated and requires a user to understand the query language.

5 Bioinformatics

Bioinformatics is a very broad field. Here I discuss only the topics that are related to my research. The core of my research is a domain-specific extraction ontology. Therefore I introduce several well known ontologies in the biology domain. I also discuss source discovery: automatically locating a proper data source and discovering its capabilities and the type of data it contains. Finally, I discuss the data trustworthiness and provenance.

The Gene Ontology (GO) [ABB+00] is a generally respected tool that provides a controlled vocabulary for the description of cellular components, molecular functions, and biological processes. GO contains 1458 components, 7413 functions, and 8907 process terms as of September 20, 2004. Many model organism databases and genome annotation groups use the GO and contribute their annotation sets to the GO resource. The goal of GO is to produce a controlled vocabulary that can be applied to all organisms even as knowledge of gene and protein roles in cells is accumulating and changing. GO provides three structured networks of defined terms to describe gene product attributes. Each node in the network also provides a connection to many other annotated genes that have similar biological function, cellular localization, or molecular process. GO is not an extraction ontology. It does not provide value recognizers (although it can be considered as a lexicon), nor does it provide general relationships among terms (although it does provide taxonomic

relationships).

LinKBase [VSD+03] is a proprietary biomedical ontology that has been developed for the purpose of making computers understand medical natural language. It comprehends various aspects of medicine, such as anatomy, diseases, and pharmaceuticals. The ontology contains 543 different relations (link types), divided into different groups, including spatial, temporal and process-related link types. LinKBase currently contains over 2,000,000 medical concepts organized in a graph with over 5,300,000 link-type instantiations. Both concepts and links are language independent, but they are cross-referenced to about 3,000,000 terms in various languages. LinKBase has been expanded by taking concepts from the Gene Ontology and virtually expanded by including mappings from the protein database Swiss-Prot to the biomedical ontology. The mapping procedure is semi-automatic. LinKBase only tries to map the top-layer concepts and GO terms with the existing LinKBase concepts/terms. Each of the three GO sub-domains, however, contains dozens of layers in its hierarchical structured vocabulary list. Therefore, it is possible to miss many matches if LinKBase only maps the top-layer concepts. Furthermore, LinKBase can only describe several binary relationships, such as "is-a" and "has-function." Biological information, however, contains many complicated relationships that LinKBase cannot cover.

Finding appropriate web resources is the first step in integrating biological data or answering users' queries. [RC03] introduces a system for finding classes of bioinformatics data sources automatically. This approach first pre-defines different classes of web sources using class descriptions. A class description presents the relevant aspects of the class from the perspective of an external application. The description considers the data type(s), example queries, outputs, and a control flow representing how types can interact with a web source. It then groups web sources into classes that share common feature. The class descriptions in this approach, however, are not generated automatically. Therefore, it is not easy to add new sources and this approach is not robust to change.

For scientific data, such as biological data, it is important to keep track of the quality of the data: the trustworthiness and the provenance of the data. Trustworthiness depends on the consistency, reliability, competence, and honesty of the data source. Provenance tells where something originated or was nurtured in its early existence. In bioinformatics research, since data has been collected from different sources, it is important to record a history of the sources, transformations, annotations, and updates of each piece of data [BCC+02].

6 Summary

In this background research study, I introduced several research areas that are closely related to my research. For each of the research areas, I surveyed different approaches that represent major ways of solving the same or similar problems. Although these approaches have made certain contributions, there are still more improvements that need to be made. In my research, I plan to develop a system that overcomes some of the drawbacks of the existing approaches and elaborate new algorithms to solve the problem of locating and extracting data from heterogenous biological sources.

References

[ABB+00] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

[BCC+02] D. Buttler, M. Coleman, T. Critchlow, R. Fileto, W. Han, C. Pu, D. Rocco, and L. Xiong. Querying multiple bioinformatics information sources: can semantic web research help? *SIGMOD Record*, 31(4):59–64, 2002.

[BFG01] R. Baumgartner, S. Flesca, and G. Gottlob. Visual Web information extraction with Lixto. In *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB'01)*, pages 119–128, Rome, Italy, September 2001.

[BLHL01] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, pages 28–37, May 2001.

[CM99] M.E. Califf and R.J. Mooney. Relational learning of pattern-match rules for information extraction. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, pages 487–493, Orlando, Florida, July 1999.

[CMM01] V. Crescenzi, G. Mecca, and P. Merialdo. Roadrunner: Towards automatic data extraction from large web sites. In *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB'01)*, pages 109–118, Rome, Italy, September 2001.

[DDH01] A. Doan, P. Domingos, and A. Halevy. Reconciling schemas of disparate data sources: A machine-learning approach. In *Proceedings of the 2001 ACM SIG-*

MOD International Conference on Management of Data, pages 509–520, Santa Barbara, California, May 2001.

[DK03] S. Decker and V. Kashyap. The semantic web: semantics for data on the web. In *Proceedings of the 29th International Conference on Very Large Data Bases (VLDB'03)*, Berlin, Germany, September 2003.

[DMD+03] AnHai Doan, Jayant Madhavan, Robin Dhamankar, Pedro Domingos, and Alon Y. Halevy. Learning to match ontologies on the semantic web. *The VLDB Journal*, 12(4):303–319, 2003.

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

[DR02] H. Do and E. Rahm. COMA—a system for flexible combination of schema matching approaches. In *Proceedings of the 28th International Conference on Very Large Databases (VLDB'02)*, pages 610–621, Hong Kong, China, August 2002.

[EA96] T. Etzold and P. Argos. SRS: Information retrieval system for molecular biology data banks. *Methods Enzymol*, 266:114–128, 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.

[Fre98] D. Freitag. Information extraction from HTML: Application of a general machine learning approach. In *Proceedings Fourteenth National Conference on Artificial Intelligence (AAAI-1998) / the Tenth Innovative Applications of Artificial Intelligence Conference (IAAI-1998)*, pages 517–523, Madison, Wisconsin, 1998.

[GDAW03] R. Gaizauskas, G. Demetriou, P. J. Artymiuk, and P. Willett. Protein structures and information extraction from biological texts: The PASTA system. *Bioinformatics*, 19(1):135–143, 2003.

[Haa01] L.M. Haas. DiscoveryLink: A system for integrated access to life sciences data sources. *IBM Systems Journal*, 40(2):489–511, 2001.

[HK04] Thomas Hernandez and Subbarao Kambhampati. Integration of biological sources: Current systems and challenges ahead. *SIGMOD Record*, 33(3):51–60, September 2004.

[HPSH03] I. Horrocks, Peter F. Patel-Schneider, and R. V. Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7, 2003.

[KPT+04] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff. Semantic annotation, indexing, and retrieval. *Journal of Web Semantics*, 2(1):49–79, 2004.

[KRMF00] Michael Krauthammer, Andrey Rzhetsky, Pavel Morozov, and Carol Friedman. Using BLAST for identifying gene and protein names in journal articles. *Gene*, 259(1-2):245–252, 2000.

[MBR01] J. Madhavan, P.A. Bernstein, and E. Rahm. Generic schema matching with Cupid. In *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB'01)*, pages 49–58, Rome, Italy, September 2001.

[RB01] E. Rahm and P.A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10:334–350, 2001.

[RC03] D. Rocco and T. Critchlow. Automatic discovery and classification of bioinformatics web sources. *Bioinformatics*, 19(15):1927–1933, 2003.

[RGM01] S. Raghavan and H. Garcia-Molina. Crawling the hidden web. In *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB'01)*, pages 129–138, Rome, Italy, September 2001.

[SBB+00] R. Stevens, P. Baker, S. Bechhofer, G. Ng, A. Jacoby, N. W. Paton, C. A. Goble, and A. Brass. TAMBIS: transparent access to multiple bioinformatics information sources. *Bioinformatics*, 16(2):184–185, 2000.

[VSD+03] J.-L. Verschelde, M. C. Santos, T. Deray, B. Smith, and W. Ceusters. Ontology-assisted database integration to support natural language processing and biomedical data-mining. In *Symposium on Integrative Bioinformatics*, Bielefeld, Germany, August 2003.