Optimized Seamless Integration of Biomolecular Data

Barbara A. Eckman* Zoé Lacroix† Louiqa Raschid‡

Abstract

Today, scientific data is inevitably digitized, stored in a wide variety of heterogeneous formats, and is accessible over the Internet. Scientists need to access an integrated view of multiple remote or local heterogeneous data sources. They then integrate the results of complex queries and apply further analysis and visualization to support the task of scientific discovery. Building such a digital library for scientific discovery requires accessing and manipulating data extracted from flat files or databases, documents retrieved from the Web, as well as data that is locally materialized in warehouses or is generated by software. We consider several tasks to provide optimized and seamless integration of biomolecular data. Challenges to be addressed include capturing and representing source capabilities; developing a methodology to acquire and represent semantic knowledge and metadata about source contents, overlap in source contents, and access costs; and decision support to select sources and capabilities using cost based and semantic knowledge, and generating low cost query evaluation plans.

1 Introduction

Scientists today spend significant time and effort in querying multiple remote or local heterogeneous data sources, and integrating the results, either manually, or with the aid of data integration tools, so that they may be further manipulated using advanced data analysis and visualization tools. In each specific application domain, getting access to, and combining data from, multiple data sources, while coping with their distribution and heterogeneity, is a tremendously difficult task.

An important aspect of bioinformatics consists in building a scientific digital library, representing an (integrated) view of data of interest. This data may be widely distributed in remote sources that are being constantly updated or it may be in local collections in data warehouses. Biological data is available in a wide variety of formats, it is annotated using a variety of methods, and it is stored in either flat files or relational or object-oriented databases. Access to these heterogeneous biological data sources is mandatory to the task of scientific discovery. A single query may involve flat files such as GenBank [BKML$^+$00] or SwissProt [BA99], Web resources such as GeneCards [RCCPL98, Genb], UniGene [Uni], or the references data source PubMed [Pub]. Result structures for Web data sources vary from loosely structured HTML format for GeneCards, to fully structured XML format for all National Center for Biotechnology Information (NCBI) Web data sources such as PubMed, to ASN.1 data exchange format. As is common with other data sources in scientific domains, biological data sources may not always support a standard programming interface (an API with a set of methods), or a standard query language, e.g., SQL, to access these sources. Typically, they support a wide range of useful tools such as keyword (text) based search engines, similarity search and sequence comparison tools usch as BLAST [AGM$^+$90] or LASSAP [GJJ97], as well as forms based interfaces and their underlying scripts.

Research in architectures and tools for data integration has been extensively investigated in the database community [Wie92]. Approaches that have been successfully developed include materialization of data in warehouses, middleware solutions to facilitate interoperability and data exchange, and heterogeneous distributed DBMS or mediation techniques that facilitate data integration.

Wrappers [BGRV99, CHN$^+$95, RS97, SA99, CDSS98, Lac00] provide tools to access remote data sources and to translate / transform the results into some common integrated representation. Data warehouses often use wrappers to import data from remote sources that is then materialized locally, and queries are evaluated against the warehoused data. Mediators and heterogeneous DBMS, on the other hand, submit queries to wrappers, and integrate the results locally to provide answers to queries. There are advantages to both approaches, as has been discussed in [EKL01]. A key disadvantage of the warehouse approach is the need for local administrators to maintain the data, while a key advantage is the control that it provides over the contents of the warehoused data. Our approach is based on mediation as will be described in the next section.

Several systems have been designed for domain specific integration of biomolecular data, providing non-materialized views of biological data sources. They include BioKleisli [DOTW97, BCD$^+$98] and its extensions K2 [DCB$^+$01] and Pizzkell/Kleisli [Won00], the TINet multi-database system based on the Object Protocol Model (OPM) and its Object-Web Wrapper [EKL01, Lac00], DiscoveryLink [HKR$^+$00], an extension for life sciences of DataJoiner and DB2 [Cha98] merged with Garlic [CHN$^+$95], P/FDM [KRG99, KAG00] and TAMBIS [BBB$^+$98]. We briefly review the features of these systems and then present the challenges that remain to be addressed.

BioKleisli follows a mediation approach and enables queries against integrated data sources [Won98, Won00]. P/FDM provides support to access specific capabilities of sources such as SRS [EA93]. Both solutions are limited in that they provide restricted access to the data sources. Queries on these sources are not optimized and typically follow a pre-determined execution plan. We will show examples of how our approach provides flexible access to data sources. TAMBIS, which uses BioKleisli, is primarily concerned with overcoming semantic heterogeneity through the use of ontologies. It provides an integrated view of data sources but offers no ad hoc interface to utilize the information retrieval tools that are available at each source. Thus, it too restricts the extent to which sources can be exploited.

DiscoveryLink [HKR$^+$00] encapsulates the access to specialized search capabilities into user-defined functions. The underlying Garlic mediator makes extensive use of cost based information to optimize access to the data sources. However, semantic knowledge about sources and query capabilities are not as yet fully utilized in the query planning phase, when sources are selected. This issue will be discussed further in a later section. The OPM multi-database system is based on the Object Protocol Model (OPM) [CM95]. OPM and an OQL-like query language are used to design object views [CKMS97] of the sources. It has a convenient architecture and APIs for its extension to wrappers, which are CORBA servers. Tools such as BLAST may be wrapped through an OPM class called Application Specific Data Type (ASDT) [TKM99]. While OPM provides the ability to evaluate complex queries, it too does not address the issue of efficient access to these sources.

To summarize, these systems have made many inroads into the task of data integration from diverse data sources. However, there remain three significant challenges that must be addressed if scientists are to be provided transparent and efficient access to diverse biological data sources to facilitate the task of scientific discovery. We address these challenges in this paper.

The first challenge is adequately capturing the diverse and often complex query capabilities of these sources, and specifying them in a catalog representation that can be used during both query formulation and query evaluation. While previous research has addressed capabilities [BGL$^+$99, LRO96, PV99, Vid00, VP97, YLGMU99], they have not addressed the diverse capabilities of biological sources. The W3C Semantic Web Activity [BLCS99, BLHL01] aims to provide a metadata layer to permit people and applications to share data on the Web. Recent efforts within the bioinformatics community address the use of OIL [FHvH$^+$00] to capture alternative representations of data to extend biomolecular ontologies [Cri00]. Such efforts focus on data representation of the contents of the sources. In contrast, we are interested in representing the actual (possibly complex) query capabilities of sources; the aspect of data representation is just one part of this challenge.

The second challenge is that there are few tools or methodologies available to examine the contents of the sources and to extract and represent metadata about the sources. Useful metadata includes domain definitions; metrics such as end-to-end latencies, cardinalities and distribution of values for some domain; as well as semantic knowledge about the contents of sources and relationships among multiple sources. As we illustrate using examples, there is significant overlap in content among the sources in this domain, as well as innumerable links between sources.

Finally, there has been research on representing the query capability of sources, and using these capabilities to reformulate queries, or capability based rewriting (CBR). There also has been some work on optimization using these capabilities. These techniques have been implemented in mediator systems [LRO96, FLMS99, HKWY97, NGT98, ROL99, YLGMU99, VRG98, Vid00, NK01]. This research must be extended to exploit the complex and diverse capabilities and metadata of biological data sources. Our objective is to use semantic knowledge about sources and query capabilities, and knowledge of query evaluation costs, to provide semantics and cost based decision support to select sources and query capabilities, and to generate low cost query evaluation plans (plans) in an efficient manner. We note that DiscoveryLink [HSK$^+$01] is the only system that provides cost based query optimization and we compare their approach with our work in a later section.

This paper is organized as follows: Section 2 describes biological data sources, capabilities and metadata. Section 3 describes query evaluation within a mediator. We discuss specification of source capabilities and metadata, and illustrate the task of generating low cost plans using example queries. Section 4 then considers issues in query optimization. Section 5 concludes.

2 Biological Data Sources, Capabilities and Metadata

There exist thousands of public data sources: there are over 110 genetics databases, and 226 relevant resources in molecular biology alone [Bax00], etc. Many data sources contain large amounts of data: For example, as of Jan 1 2001, GenBank [BKML$^+$00] contained 11,101,066,288 bases in 10,106,023 sequences, and its growth continues to be exponential, doubling every 14 months [Gena]. While the number of distinct human genes appears to be smaller than expected, in the range of 30-40,000 [Con01, V$^+$01], the distinct human proteins in the proteome are expected to number in the millions, due to the apparent frequency of alternative splicing, RNA editing, and post-translational modification [CSC$^+$00, Fra01, Gra01]. The discovery and annotation of interactions among these proteins represents a significant challenge to current bioinformatics data management and integration tools.

Access to multiple heterogeneous biological data sources is mandatory to the task of scientific discovery. A single query may involve flat files (that may be stored locally) such as GenBank [BKML$^+$00] or SwissProt [BA99], Web resources such as GeneCards [RCCPL98, Genb], UniGene [Uni], or the references data source PubMed [Pub]. These sources are textual and provide limited query capability. Their data structure varies from loosely structured HTML format for GeneCards to fully structured XML format for all National Center for Biotechnology Information (NCBI) Web data sources such as PubMed, to the ASN.1 data exchange format. As is common with other data sources in scientific domains, biological data sources may not always support a standard API (set of methods), or a standard query language, e.g., SQL, to access these sources. Typically, they support a wide range of useful tools such as keyword (text) based search engines, similarity search and sequence comparison tools usch as BLAST [AGM$^+$90] or LASSAP [GJJ97], and forms based interfaces and their underlying scripts.

Most of the biological data sources publicly available on the Web provide browsing capabilities. For example, given a HUGO name [WJ97], a biologist can retrieve a summary of significant information about a gene from GeneCards, then click on the UniGene cluster and access relevant information from UniGene, and then click again to access GenBank. From there, one can access PubMed by clicking on the MEDLINE identifier, and associated references may be retrieved.

While a link-driven federation of Web biological data sources providing *browsing* capabilities is very useful for scientific discovery, it does not provide sophisticated support to the scientist to relieve the tedium involved in *querying* multiple data sources, following multiple links from each data source, and extracting and integrating the relevant data for further investigation. We illustrate the differences between browsing and querying using examples.

First a source may not directly support the complete navigation capability that is implicitly captured by its contents. For example, a MEDLINE-formatted PubMed citation provides a GenBank or SwissProt identifier (in ASCII), without always providing the hyperlinks to access the corresponding data sources. To navigate from PubMed to GenBank (or SwissProt), a user must extract the corresponding parameters, then construct a call (in the browser), and finally submit a query to the next source. In contrast, a mediator based data integration solution can provide seamless access to both the navigation capabilities directly implemented by the data provider, as well as the capabilities that require submitting calls to the remote sources. We note that the Entrez interface to PubMed does support an additional capability associated with the Nucleotide Link that provides access to GenBank entries. This is discussed in more detail in a later section.

Second, scientific discovery typically requires evaluating complex queries on multiple sources [EKL01]. Discovering the often complex query capabilities of these multiple sources can be a daunting task. Equally difficult is providing efficient access to the multiple sources. We discuss both challenges using the following example:

> **Query 1:** *Return all citations of PubMed published since 1995 that mention "heart" and refer to sequences of GenBank that are annotated as "calcium channel".*

The task of scientific discovery encapsulated by this query would not be possible without knowledge of the capabilities of sources. For example, PubMed and GenBank accept queries containing multiple selection predicates. This includes either PubMedIds or GenBankIds, which support database like selection, as well as keywords that correspond to a more sophisticated search capability. Constraints such as `date since 1995` can also be used to filter the results. This increases the query capability and can improve efficiency of access. Both the keyword `brain` and the constraint `date since 1995` can be passed to PubMed to filter the results. PubMed typically removes duplicates when they occur in the output. However, sending a single request, even if the output contains duplicates that must be eliminated locally, may be less expensive than submitting multiple calls on the Web.

While knowledge of the capabilities provides some insight into efficient access, as discussed in this example, as the complexity of the queries increase, there are many potential combinations of accesses to evaluate a query. Each combination is a query evaluation plan or plan for the query. Each plan will have a different cost. Choosing among plans to obtain a low cost plan, or the task of optimization, can be very difficult, and we illustrate using the above example.

To answer Query 1, one can access PubMed and retrieve citations published since 1995 that mention "heart", and then extract all GenBank identifiers that they contain. Next, one can retrieve the information available in GenBank for each sequence and filter the ones that are annotated as "calcium channel".

A different plan would first access GenBank and retrieve all sequences that are annotated as "calcium channel", and then extract the MedLine identifiers for these. Next one can retrieve all MedLine citations from PubMed and filter the ones published since 1995 that mention "heart".

The two possible plans to evaluate the same query are far from being identical. We first consider evaluation costs for executing the plans. Each access to a data source retrieves many documents that need to be parsed. Each object that is returned may generate further accesses to (other) sources. Web accesses are costly and should be as limited as possible. To limit the number of calls, there is a need to examine plans that are selective as early as possible. For example, the call to PubMed in the first plan retrieves 81,840 citations, whereas the call to GenBank in the second plan retrieves 1,616 sequences. If each of the retrieved documents (from PubMed or GenBank) generated an additional access to the second source, then clearly the second plan has the potential to be much less expensive when compared to the first plan.

Information about source capabilities and domain specific metadata is needed to identify the differing evaluation costs of alternate plans. Both PubMed and GenBank are very large databases. GenBank contains over 10 million sequences [Gena]. PubMed accesses citations that include the over 11 million article references registered in MedLine. Many GenBank entries are unpublished or are registered as US patents, and therefore are not associated with PubMed entries. On the other hand, most citations retrieved from PubMed do not contain GenBank entries.

While determining the evaluation costs of plans is important, there is an even more critical issue that must be explored. This is determining if the alternate plans indeed produce identical answers to the query, and are *semantically equivalent* as defined by the standards of some community, or if the alternate plans produce non-equivalent answers. We note that knowledge to make this determination is non trivial and requires extensive domain expertise.

To explain further, while our approach to specifying query capabilities and metadata is capable of identifying alternate sources and capabilities, there is an additional significant task to address: semantic data integration. First, similar or related information may be expressed using different formats, representation models or equivalent vocabulary. Further, while two query capabilities may be similar, they are not always semantically equivalent; informally, they may not provide identical answers. To explain, there are explicit occurrences of GenBank entries embedded within the MEDLINE formatted description of a citation entry in PubMed. In addition, the Nucleotide Links display option also provides links to GenBank entries. From the viewpoint of query processing capabilities, we cannot distinguish between these two capabilities, since they both accept a PubMed citation as input and return GenBank identifiers as output. However, the semantics associated with these two evaluation plans is completely different. For example, the citation identified in PubMed as 8552191 refers explicitly to 4 GenBank identifiers in its MEDLINE representation, whereas the Nucleotide Link returns 8 GenBank identifiers. (Note: The queries were executed in August 2001.)

The two alternate plans that were described to evaluate Query 1 may also produce answers which are not identical. Our approach will provide the ability to express the metadata to capture important semantic knowledge about data sources, and to use this knowledge to identify plans that are more efficient as well as semantically equivalent.

In the next section, we discuss the representation of query capability and metadata. We then use queries to illustrate alternate plans with different costs. Finally, we consider some issues in optimization that are of importance for biological data sources.

3 Mediator Query Evaluation

Figure 1 presents the components of a wrapper mediator architecture tailored to access heterogeneous information sources (WebSources) across wide area networks. Research in mediators [Wie92] is reported in [BGL+99, LRO96, FLMS99, HKWY97, NGT98, ROL99, TRV98, YLGMU99, VRG98, Vid00]. The mediator has the task of decomposing a mediator query into subqueries that are executed on individual sources. The mediator identifies relevant sources that can answer each subquery. The mediator also provides query optimization and evaluation functionalities for the mediator query. Query planning and optimization techniques tailored to optimize mediator queries on biological sources will be discussed in the next section.

The mediator references a Catalog that has metadata about source capabilities and source contents. Useful metadata includes domain definitions; metrics such as end-to-end latencies, cardinalities and distribution of values for some domain; and semantic knowledge about the contents of sources and relationships among multiple sources.

A Wrapper Broker interoperates between the mediator and wrappers. Wrappers [BGRV99, CHN+95, RS97, SA99, CDSS98, Lac00] handle query execution of wrapper subqueries on individual sources. A source is accessible via the `http` protocol, and stores data in database or non database servers. A Web-based interface provides a limited query capability, and returns answers in XML or HTML or ASCII. Results of wrapper subqueries that are returned by individual wrappers are further processed by the Evaluation Engine of the mediator. If the wrapper

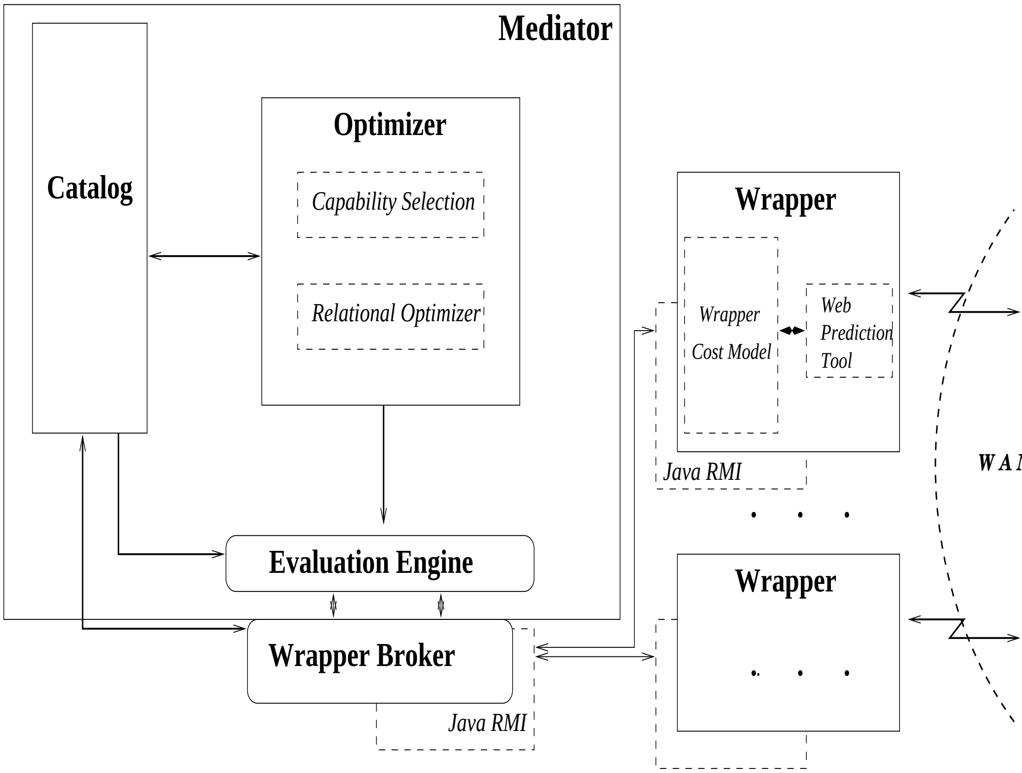

Figure 1: Wrapper Mediator Architecture

returns ASCII or HTML, it is first converted to XML. Next, the XML is processed by XML queries [CFR+00] that can express all expected data transformations (reorganization) and manipulations as in [Lac01].

Next, we describe how source capabilities may be formally represented by the mediator. We then illustrate plans for some sample queries.

3.1 Source Capabilities

We refer those readers who are not familiar with the biological data sources that have been mentioned to Figure 2 that identifies some of the commonly accessed biological data sources. The purpose of the figure is to simply illustrate the navigation capabilities that are provided. We note that it is not meant to represent accurate details of specific sources and their capabilities. In the rest of this section, we discuss how specific source capabilities can be accurately described.

Each of the data sources is represented by a mediator schema. For simplicity, we assume the use of the relational data model for the mediator schema. We also assume that each source, e.g. PubMed, implements a single relation PubMed. It is the task of the PubMed wrapper to provide an interface which accepts queries on the PubMed relation; process these queries against the actual contents of the PubMed source; and return appropriate results.

We note that the sources typically may not support the mediator schema, e.g., the PubMed source may not actually implement a relation PubMed. We further note that most sources are semi-structured; their contents would typically be represented by multiple object types (classes); and a complex result (object) from one or more of these classes could be encapsulated within the answers.

We describe the query capability of a source as an *input-output* relationship *ior* of the form *ior*: $Input \rightarrow Output$ or *ior*: $Input \rightarrow\rightarrow Output$; the *ior* is further described in Figures 3, 4 and 5.

Most sources do not support arbitrary select-project-join (SPJ) queries over their content, e.g., PubMed will

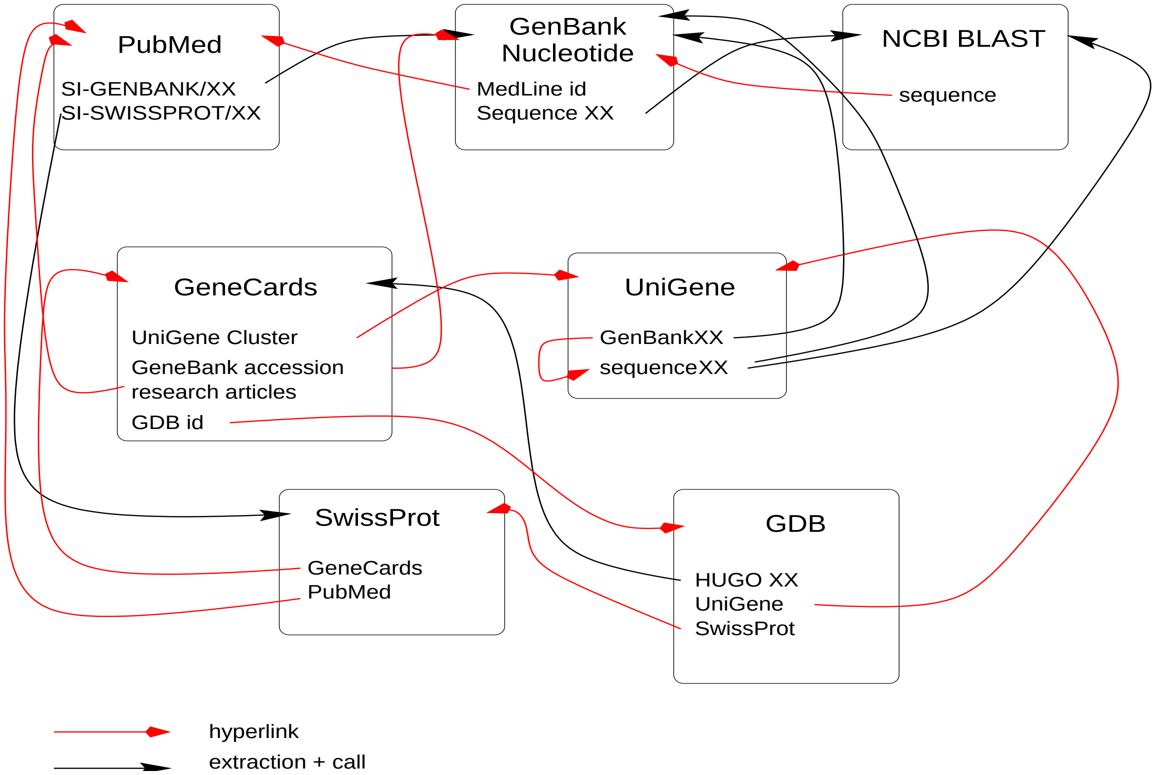

Figure 2: Web data sources capabilities

not support a scan over relation PubMed that will return all the citation records that it maintains. Thus, an *ior* specifies the actual query capability implemented by the source. Typically, the *Input* is a set of bindings for input attributes. The *Output* are the elements that are returned in the answer. The relationship between *Input* and *Output* could be represented by a functional or a multi-valued dependency.

The values for *Input* may be singular or set valued; they also may be required or optional (see Figure 4). The types and occurrences of *Output* are in Figure 5. To explain, the output could be (scalar) values or hyperlinks. A *global identifier* is a unique identifier (key) maintained by a particular source, e.g., `MedLineId`. The values of the output could also be singular or set valued. In the case that there is a multi-valued dependency between the *Input* and the *Output*, there may be duplicates occurring in the output. {Dup|NoDup} indicates if these duplicates are eliminated. Most of the sources are semi-structured; missing values for attributes in the output may not be interpreted as NULL (as is done in the relational model). We distinguish the relational model from the semi-structured model with *All* and *All* *, respectively.

Figure 6 illustrates some of the capabilities implemented by the sources that we have mentioned. For example, capability ior_cit1 of source PubMed accepts a set of values of input bindings of `PubMedId`. This capability (ior_cit1) also implements a functional dependency. For *each* value of `PubMedId` that is input, only one result is returned, i.e., the output is functionally dependent on each input value of `PubMedId`. In contrast, ior_cit5 implements a multi-valued dependency; this capability represents a (boolean) keyword search capability.

3.2 Examples of Queries and Plans

We use examples of queries and query evaluation plans or plans to illustrate how different plans may change the order of access to sources, or may access completely different sources to answer a query.

3.2.1 Exploring Alternate Order of Accessing Sources

> **Query 2:** *Return accession numbers and definitions of GenBank EST sequences that are similar (60% Identical over 50AA) to calcium channel sequences in SwissProt that have references published since 1995 and mention brain.* [EKL01]

We consider two plans that access the same sources. *However, the order of accessing the sources is different, as are the queries that are submitted to the sources.* One possible plan for this query is as follows: First access PubMed and retrieve references published since 1995 that mention "brain". Then extract from all these references the SwissProtId and obtain the corresponding sequences from SwissProt whose function is calcium channel. Finally, execute a BLAST search using a wrapped BLAST application to retrieve similar sequences from GenBank (gbest sequences).

An alternative approach accesses SwissProt and retrieves sequences whose function is "calcium channel". In parallel, one can retrieve citations from PubMed that mention "brain" and are published since 1995 and extract sequences from them. One can then determine which sequences are in common. Finally, one can execute a BLAST search to retrieve similar sequences from GenBank (gbest sequences).

A graphical representation of both plans is in Figures 7 and 8.

We also represent the first plan using the following algebraic notation similar to the relational algebra.

Plan 1 for Query 2

$\mathrm{T}_1 \leftarrow \pi_{SwissProtId,\cdots}\ (\mathrm{EXT}\sigma_{(brain,1995--)}(\mathrm{PubMed}))$

$\mathrm{T}_2 \leftarrow \pi_{Sequence,\cdots}\ (\mathrm{T}_1\ \mathit{Dep} \bowtie_{SwissProtId}\ \mathrm{EXT}\ \sigma_{(Ca\ Channel)}\ (\mathrm{SwissProt}))$

$\mathrm{T}_3 \leftarrow \mathrm{T}_2\ \mathit{Dep} \bowtie_{Sequence}\ \mathrm{EXT}\sigma_{(60\%,50AA)}\ (\mathrm{BLAST})$

$\mathrm{Result} \leftarrow \pi_{Accession,Definition}\ (\mathrm{T}_3)$

To explain this plan, the expression $\mathrm{EXT}\sigma_{(}\cdots)$ (PubMed) represents an external scan (plus optional selection) that is submitted to the wrapper to access data from the relations that are maintained by the source; in this case it is relation Pubmed. The actual query that is submitted to the source is identified by the *ior* which is a parameter of EXTσ. The *ior* is not identified in the plan; instead for simplicity we indicate the bindings that are passed to each EXTσ. Thus, the first source that is accessed is PubMed (using *ior* ior_cit5), and the keyword `brain` and date `since 1995` are pushed to that source. We use the operator π to identify attributes that must be extracted or projected from the results of the EXTσ operator. These attributes are either required in the final answer or it is used at intermediate nodes. The *Dep* $\bowtie$ is a dependent join operator. It is used when output from accessing one source is passed as an input binding to another source. For example, to explain the expression

$(\mathrm{T}_1\ \mathit{Dep} \bowtie_{SwissProtId}\ \mathrm{EXT}\sigma_{(CaChannel)}\ (\mathrm{SwissProt}))$,

the values of SwissProtId are obtained from evaluating the expression T_1; they are then passed via the *Dep* $\bowtie$ operator to the scan operator EXTσ that is evaluated on SwissProt. In addition, the keyword `Ca Channel` is also passed to the source SwissProt via the scan operator EXTσ.

An alternate plan for this query is as follows:

Plan 2 for Query 2

$\mathrm{T}_1 \leftarrow \pi_{Sequence,PubMedId,SwissProtId}\ (\mathrm{EXT}\ \sigma_{(CaChannel)}\ (\mathrm{SwissProt})\ )$

$\mathrm{T}_2 \leftarrow \pi_{SwissProtId}\ (\ \mathrm{EXT}\sigma_{(brain,1995--)}\ (\mathrm{PubMed})\ )$

$\mathrm{T}_3 \leftarrow \mathrm{T}_1 \bowtie_{SwissProtId} \mathrm{T}_2$ where $\bowtie$ is over SwissProtId

$\mathrm{T}_4 \leftarrow \mathrm{T}_3\ \mathit{Dep} \bowtie_{SwissProtId}\ \mathrm{EXT}\sigma_{(60\%,50AA)}\ (\mathrm{BLAST})$

In the second plan, both PubMed and SwissProt are accessed in parallel. We then perform the $\bowtie$ operator over the values of SwissProtId to determined which sequences are in common. Finally, we execute BLAST search to determine the similar EST sequences.

Verifying if the two plans are semantically equivalent, i.e., the answers that are returned from the two plans are identical, is non trivial. Metadata on the particular query capability, encapsulated by the specific *ior* used to access each source is relevant. In addition, semantic knowledge about the links between sources, and about combinations of *ior*s may also be required. The task is somewhat simplified in this query since both plans access the same sources, but in a different order.

3.2.2 Accessing Multiple Alternate Sources

Query 3: Return the UniGene cluster(s) of all SwissProt proteins with keyword apoptosis.

There are at least two potential plans for the query. *The two plans access different sources.* In the first plan, one can first access the SwissProt source and select all proteins that are annotated with the keyword `apoptosis`, and project EMBL/GenBank cross references, or the GenBankId. Next, one can access UniGene with the GenBankId and retrieve the corresponding UniGene cluster(s). The second plan will also access SwissProt. However, it will project HUGO names from this source. Next, it will retrieve the corresponding UniGene cluster(s) from the GeneCards source.

Plan 1 for Query 3

$\text{Alt\_T}_{3,1,1} \leftarrow \pi_{GenBankId}$ (EXT$\sigma_{(apoptosis)}$ (SwissProt))

$\text{Alt\_T}_{3,1,2} \leftarrow \pi_{UniGeneCl}$ ($\text{Alt\_T}_{3,1,1}$ *Dep* $\bowtie_{GenBankId}$ EXT σ (UniGene))

Plan 2 for Query 3

$\text{Alt\_T}_{3,2,1} \leftarrow \pi_{HUGO}$ (EXT$\sigma_{(apoptosis)}$ (SwissProt))

$\text{Alt\_T}_{3,2,2} \leftarrow \pi_{UniGeneCl}$ ($\text{Alt\_T}_{3,2,2}$ *Dep* $\bowtie_{HUGO}$ EXT σ (GeneCards))

We note that determining if these two plans are semantically equivalent in this example is more complex since the two plans access different sources and use significantly different capabilities. Semantic knowledge about the contents of the sources as well as the links between the sources will be required.

4 Optimization: Choosing Low Cost Plans

We first describe the task of optimization with databases. Next, we discuss the challenges of optimization with remote biological data sources. We then describe an approach that we have developed for two phase optimization. In the first pre-optimization phase, sources and capabilities are selected. In the second phase, an optimal plan is generated a la relational optimization. Finally, we discuss a decision support model for the pre-optimization phase which considers both semantics and costs. We compare our approach with related approaches, in particular the cost-based optimization of [HSK^{+}01].

4.1 Overview of Optimization

Consider a typical database query on a set of relations (or collections of objects). We can deconstruct this query as a tree of (algebraic) operators to be executed on the relations or collections of objects. Examples of relational operators are a *scan* or a *selection* operator. A *query execution plan* or plan must select an ordering of these operators. It must also select an implementation for each of these operators, i.e., a piece of algorithmic code to implement each of these operators. For example, the relational *join* operator may be implemented as a *hash join* or a *nested loop join*. A plan is then evaluated by a database evaluation engine to produce answers for a query.

Query optimization [Ull89, Ull97] is the science and the art of applying equivalence rules to rewrite the tree of operators and produce an *optimal* plan. There are well known syntactic, logical, and semantic equivalence rules that are used during optimization [Ull89]. The objective of the task of query optimization is producing an optimal or a good *low cost* query execution plan for a query. In order to obtain the cost of a plan, one must have a Catalog of accurate metrics, e.g., the cardinality or the number of result tuples in the output of each operator, the cost of accessing a source and obtaining results from that source, etc. One must also have a cost formula that can be used to calculate the processing cost for each implementation of each operator. The *overall cost* is typically defined as the total time needed to evaluate the query and obtain all of the answers.

It has been shown that the complexity of producing an optimal low cost plan for a relational query is NP-complete [Mor88, Ull89], and there is much work on developing reasonable heuristics to solve this problem. Both dynamic programming and randomized optimization based on simulated annealing provide good solutions [IK90, SAC$^+$79, SMK97, UFA98].

Next, we consider the special challenges of query evaluation on remote biological data sources. There has been significant research on query optimization for sources with multiple diverse capabilities in recent years. This extension to query optimization has been termed *capability based rewriting* or CBR, to reflect that the rewriting must respect the capabilities or the functionality supported by the sources. [FLM98, FLMS99, HKWY97, LRO96, TRV98, Ull97, VP97, Vid00, Wie92, YLGMU99]. It has been shown that the *size of the search space* of query execution plans increases, with multiple sources, each with different and diverse capabilities. The size of the search space also increases due to the possible overlap of content and capabilities among the sources [Vid00]. Finally, developing costs models and cost formulas for remote autonomous sources accessible on the dynamic Internet can be very difficult and there has been some effort in recent years to develop cost models [HKWY97, GRZZ00, ZRZB01, NGT98, NK01].

4.2 Two-Phase Optimization

We have developed an approach for efficient two-phased query optimization with limited capability sources [Vid00]. This approach can be briefly summarized as follows: In the first pre-optimization phase, we select sources and capabilities prior to the more expensive optimization step. A cost based decision support model is used in the pre-optimization phase to choose good *low cost* sources and capabilities that will lead to good low cost plans in the optimization phase. In the second phase, we extend a traditional relational optimizer to generate low cost plans that respect the choice of the sources and capabilities of the pre-optimization phase.

To develop a cost based decision support model for pre-optimization, we must understand the factors that impact the efficiency of execution of complex queries posed by the genome researchers. They include the size of each relation involved in the query or the cardinality of the relation; the number of results that are returned or the selectivity of the query; the number of queries that are submitted to the sources; the order of accessing sources; etc.

As of May 4th, 2001, SwissProt contains 95,674 entries whereas PubMed contains more than 11 million citations; these are the values of cardinality for the corresponding relations. Again, we note that these relations are not actually implemented in the source and it is the responsibility of the wrapper to implement the capabilities on these relations. A query submitted to PubMed (as used in the first plan to evaluate Query 2) retrieves 727,545 references that mention “brain”. Alternately, the query retrieves 206,317 references that mention ”brain” and were published since 1995. This is the selectivity of the query. We note that the query is actually identified by a specific capability and specific value(s) of binding for input attributes.

We now consider the impact of changing the order of accessing sources. The first plan for Query 2 accesses PubMed, extracts values for SwissProtId, and then passes these values of SwissProtId to the query on SwissProt, via the *Dep* $\bowtie$ operator. Passing these values of SwissProtId has the potential to constrain the query and could reduce the number of results returned from SwissProt. The second plan submits queries to both PubMed and SwissProt in parallel. It does not pass values of SwissProtId to SwissProt; potentially more results may be returned from SwissProt. However, there is a *single* query submitted to SwissProt in the second plan. Also both sources are accessed in parallel. We further note that executing a BLAST search can take up to several hours and *generally* should be performed as infrequently as possible. These factors, as well as additional factors that we will discuss next, have an impact on the cost of each plan. Further, we need to determine if alternate plans provide identical answers.

4.3 Cost Based Decision Support for Pre-Optimization

We summarize the factors that impact the decision support model that is used during pre-optimization. Several of these factors are tailored to exploit the complex and diverse capability and metadata characterizing the biological data sources.

For each combination of sources and capabilities that are relevant to a query, the decision model will use the following factors to evaluate the choice:

- Using metadata on cardinality, distribution of distinct values in selected domains, statistics associated with particular values of bindings for input attributes, etc. we estimate the selectivity of each capability. The decision model will favor capabilities that are more selective and that minimize the size of the result.

- The decision model must consider typical delays (end-to-end response time) experienced at each source. Sources that are located in Europe and Israel, may experience significant delay, as well as be sensitive to the time of the day and the day of the week [GRZZ00].

- These two factors are not independent. For example, a more expensive call (greater delay) may be more selective. The decision model may need to consider the trade-off of these two factors in making a choice.

- The model will explore different orders for accessing multiple sources. Based on order or sequences of access and the particular capabilities, we can estimate the number of queries (wrapper calls) that will be submitted to source. In some cases, we can exploit parallel access to sources as we did in the second plan for Query 2. The decision model will choose a particular order for accessing sources that reduces the number of queries and maximizes the advantage of parallel access.

We now consider the semantic knowledge that must be exploited to determine if plans are semantically equivalent and produce identical answers. When the sources and query capabilities that are accessed are identical, and when the decision model only considers different orders for accessing the sources, then verifying that the plans are semantically equivalent is straightforward. When the sources that are accessed are identical, but the query capabilities that are utilized are different, then semantic knowledge about the *ior*s as well as about links (explicit or implicit) between these sources must be examined to make a decision. Finally, when different sources are accessed, then semantic knowledge about the contents of the sources must also be considered, together with knowledge about the *ior*s and links between sources.

Once a set of (possibly) alternate sources and capabilities are chosen, which may or may not be equivalent, the combinations of sources and capabilities will be ranked by the decision model. Then, the best combination or the top N combinations will be used to drive an optimizer to generate a low cost plan.

We now briefly compare our approach with related work in optimization. The K2 optimizer [DCB$^+$01] is an extensible rule-based optimizer. It provides algebraic simplification of complex queries, and eliminates the intermediate collection of results which could be expensive. It also simplifies and rewrites queries to group operations performed on the same data source.

DiscoveryLink [HSK$^+$01, ROL99] performs extensive syntactic and semantic simplification and cost-based optimization. For example, they identify good implementations for expensive operators, e.g., sort. They also use cost based heuristics to find optimal join orderings. As in our approach, they use both query capabilities and metadata about evaluation costs to perform optimization.

A key difference from our approach is that we explicitly represent sources, query capabilities and metadata within the mediator. The Garlic mediator[ROL99] which is used by DiscoveryLink relies on the wrapper and its cost model to determine the query capability to be used to access a wrapped source. One consequence is that the wrapper makes a decision without knowledge of the complete query and the other sources and query capabilities that may also be used in the query. Thus, the Garlic mediator may not consider some potential capabilities and plans. Our approach allow us to explore alternate combinations of sources and capabilities, e.g., the alternate sources GenBank and GeneCards, for Query 3. The reliance of the Garlic mediator on semantic knowledge captured in the wrappers may not support this flexibility. A more important drawback is that since the Garlic mediator does not explicitly capture semantic knowledge of sources and capabilities, it will be more difficult to extend their approach to determine if plans are semantically equivalent. This is critical in the domain of biological data sources, where there are multiple alternate sources and complex and diverse query capabilities.

5 Conclusion

In this paper, we consider the following challenges for evaluating complex queries on biological data sources: capturing the diverse query capabilities of sources; extracting and representing metadata on the contents of sources and relationships among sources; optimization to generate low cost query evaluation plans in an efficient manner and determining if alternate plans were semantically equivalent. In future work, we will develop an extensive Catalog of source capabilities and metadata and verify the effectiveness of our optimizer. This research significantly extends on current work on metadata representation on the Web as currently carried out by the W3C Semantic Web Activity [BLCS99, BLHL01] and more specific efforts to use the OIL [FHvH+00] to capture alternative representations of data to extend biomolecular ontologies as proposed in [Cri00]. Our Catalog captures both multiple alternate sources as well as their capabilities. This knowledge will be used by our mediator to explore multiple alternate evaluation plans to choose the best (least cost) query evaluation plan and to determine if plans are semantically equivalent. None of the current efforts to optimize query execution against Web sources addresses the variety of source capabilities of our catalog or addresses the issue of semantic equivalence.

Acknowledgments: We thank Susan Chacko of the National Institutes of Health for her feedback.

References

[AGM+90] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman. Basic Local Alignment Search Tool. *Journal of Molecular Biology*, 215(3):403–10, October 1990. http://www.ncbi.nlm.nih.gov/BLAST.

[BA99] A. Bairoch and R. Apweiler. The SWISS-PROT protein sequence databank and its supplement TrEMBL. *Nucleic Acids Res*, 1(27):49–54, January 1999. http://www.expasy.ch/sprot.

[Bax00] A. Baxevanis. The molecular biology database collection. *Nucleic Acids Research*, 28(1):1–7, 2000. http://nar.oupjournals.org/cgi/content/ full/27/1/1.

[BBB+98] P. Baker, A. Brass, S. Bechhofer, C. Goble, N. Paton, and R. Stevens. TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources. An Overview. In *Sixth International Conference on Intelligent Systems for Molecular Biology (ISBM98)*, 1998.

[BCD+98] P. Buneman, J. Crabtree, S. Davidson, V. Tannen, and L. Wong. BioKleisli. *BioInformatics*, 1998. http://www.cbil.upenn.edu/K2/.

[BGL$^+$99] C. Baru, A. Gupta, B. Ludäscher, R. Marciano, Y. Papakonstantinou, P. Velikhov, and V. Chu. XML-Based Information Mediation with MIX. In *Proc. ACM SIGMOD Symp. on the Management of Data*, pages 597–599, Philadelphia, Pennsylvania, 1999. demonstration session.

[BGRV99] L. Bright, J-R Gruser, L. Raschid, and M.E. Vidal. A wrapper generation toolkit to specify and construc wrappers for web accessible data sources (websources). *Journal of Computer Systems Science & Engineering. Special Issue: Semantics on the World Wide Web*, 14(2), March 1999.

[BKML$^+$00] D. Benson, I. Karsch-Mizrachi, D. Lipman, J. Ostell, B. Rapp, and D. Wheeler. GenBank. *Nucleic Acids Res*, 1(28):15–8, January 2000. http://www.ncbi.nlm.nih.gov/Genbank.

[BLCS99] T. Berners-Lee, D. Connoly, and R. Swick. *Web Architecture: Describing and Exchanging Data.* W3C Note, June 1999.

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

[CDSS98] S. Cluet, C. Delobel, J. Simeon, and K. Smaga. Your mediators need data conversion. In *Proccedings of the ACM SIGMOD Conference*, pages 177–188, 1998.

[CFR$^+$00] D. Chamberlin, D. Florescu, J. Robie, J.Siméon, and M. Stefanescu. *XQuery: A Query Language for XML.* W3C, 2000. available at http://www.w3.org/TR/xmlquery.

[Cha98] D. Chamberlin. 1998.

[CHN$^+$95] W.F. Cody, L.M. Haas, W. Niblack, M. Arya, M.J. Carey, R. Fagin, D. Lee, D. Petkovic, P.M. Schwarz, J. Thomas, M. Tork Roth, J.H. Williams, and E.L. Wimmers. Querying multimedia data from multiple repositories by content: The garlic project. In *IFIP 2.6 Third Working Conference on Visual Database Systems (VDB-3)*, Lausanne, Switzerland, March 1995. http://www.almaden.ibm.com/cs/garlic.

[CKMS97] I.A. Chen, A.S. Kosky, V.M. Markowitz, and E. Szeto. Constructing and Maintaining Scientific Database Views. In *Proceedings of the 9th Conference on Scientific and Statistical Database Management*, August 1997.

[CM95] I.A. Chen and V.M. Markowitz. An Overview of the Object-Protocol Model (OPM) and OPM Data Management Tools. *Information Systems*, 20(5):pp 393–418, 1995.

[Con01] International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, February 2001.

[Cri00] T. Critchlow. Report on xewa-00: The xml enabled wide-area searches for bioinformatics workshop. Technical report, IEEE Computer Society, 2000.

[CSC$^+$00] L. Croft, S. Schandorff, F. Clark, K. Burrage, P. Arctander, and J. Mattick. Isis: The intron information system reveals the high frequency of alternative splicing in the human genome. *Nature Genetics*, 24:340–341, 2000.

[DCB$^+$01] S. Davidson, J. Cabtree, B. Brunk, J. Schug, V. Tannen, C. Overton, and C. Stoeckert. K2/kleisli and gus: Experiments in integrated access to genomic data sources. *IBM Systems Journal*, 40(2):512–531, 2001.

[DOTW97] S. Davidson, C. Overton, V. Tannen, and L. Wong. BioKleisli: A Digital Library for Biomedical Researchers. *Journal of Digital Libraries*, 1997.

[EA93] T. Etzold and P. Argos. SRS, An Indexing and Retrieval Tool for Flat File Data Libraries. *Computer Applications of Biosciences*, 9(1):49–57, 1993. See also http://srs.ebi.ac.uk.

[EKL01] B. Eckman, A. Kosky, and L. Laroco. Extending traditional query-based integration approaches for functional characterization of post-genomic data. *BioInformatics*, 17(7):587–601, 2001.

[FHvH$^+$00] D. Fensel, I. Horrocks, F. van Harmelem, S. Decker, M. Erdman, and M. Klein. OIL in a Nutshell. In *Proceedings of the European Knowledge Acquisition Conference (EKAW'00).* LNAI, Springer Verlag, 2000.

[FLM98] D. Florescu, A. Levy, and A. Mendelzon. Database techniques for the world-wide web: A survey. *SIGMOD Record*, September 1998.

[FLMS99] D. Florescu, A. Levy, I. Manolescu, and D. Suciu. Query optimization in the presence of limited access patterns. In *Proccedings of the* ACM SIGMOD Conference, 1999.

[Fra01] L. Frank. Between genes and proteins: After dna, scientists face the complex world of rna. 2001. http://www.celera.com/genomics/news/articles/03_01/Between_genes_proteins.cfm".

[Gena] GenBank. Growth of genbank. (available at http://www.ncbi.nlm.nih.gov/ Genbank/genbankstats.html).

[Genb] GeneCards. http://bioinformatics.weizmann.ac.il/cards/. Weizmann Institute Genome Center and Bioinformatics Unit.

[GJJ97] E. Glemet and Codani. J.-J. Lassap: a large scale sequence comparison package. *Bioinformatics*, 13(2):137–143, 1997.

[Gra01] B. Graveley. Alternative splicing: Increasing diversity in the proteomic world. *Trends in Genetics*, 17(2):100–107, 2001.

[GRZZ00] J.R. Gruser, L. Raschid, V. Zadorozhny, and T. Zhan. Learning response time for websources using query feedback and application in query optimization. *VLDB Journal, Special Issue on Databases and the Web*, (1):18–37, 2000.

[HKR$^+$00] L. Haas, P. Kodali, J. Rice, P. Schwarz, and W. Swope. Integrating Life Sciences Data - With a Little Garlic. In *IEEE International Symposium on Bio-Informatics and Biomedical Engineering (BIBE)*, Published by IEEE Press, Washington, DC, November 2000.

[HKWY97] L. Haas, D. Kossmann, E. Wimmers, and J. Yang. Optimizing queries across diverse data sources. In *Proceedings of VLDB Conference*, 1997.

[HSK$^+$01] L. Haas, P. Schwarz, P. Kodali, E. Kotlar, J. Rice, and W. Swope. Discoverylink: A system for integrating life sciences data. *IBM Systems Journal*, 40(2), 2001.

[IK90] Y. Ioanidis and Y. Kang. Randomized algorithms for optimizing large join queries. In *Proccedings of the ACM Sigmod Conference*, 1990.

[KAG00] G. Kemp, N. Angelopoulos, and P. Gray. A Schema-based Approach to Building a Bioinformatics Database Federation. In *IEEE International Symposium on Bio-Informatics and Biomedical Engineering (BIBE)*, Published by IEEE Press, Washington, DC, November 2000.

[KRG99] G. Kemp, C. Robertson, and P. Gray. Efficient access to biological databases using CORBA. *CCP11 Newsletter*, 3.1(7), February 1999.

[Lac00] Z. Lacroix. Scientific Data Integration: Wrapping Textual Documents with a Database View Mechanism and an XML Engine. In *IEEE International Symposium on Bio-Informatics and Biomedical Engineering (BIBE)*, Published by IEEE Press, Washington, DC, November 2000.

[Lac01] Z. Lacroix. Retrieving and Extracting Web Data with Search Views and an XML Engine. In *International Workshop on Data Integration over the Web*, Interlaken, Swizterland, June 2001.

[LRO96] A. Levy, A. Rajaraman, and J. Ordille. Querying heterogeneous information sources using source descriptions. In *Proceedings of the VLDB Conference*, 1996.

[Mor88] K. Morris. An algorithm for ordering subgoals in nail! In *Proceedings of Symposium on Principles of Database Systems*, 1988.

[NGT98] H. Naacke, G. Gardarin, and A. Tomasic. Leveraging mediator cost models with heterogeneous data sources. *Proc. of ICDE*, 1998.

[NK01] Z. Nie and S. Kambhampati. Joint optimization of cost and coverage of information gathering plans. (submitted to publication), 2001.

[Pub] PubMed. http://www.ncbi.nlm.nih.gov/pubmed/. National Library of Medecine.

[PV99] Y. Papakonstantinou and V. Vassalos. Query Rewriting for Semistructured Data. In *Proc. ACM SIGMOD Symp. on the Management of Data*, pages 455–466, Philadelphia, Pennsylvania, 1999.

[RCCPL98] M. Rebhan, V. Chalifa-Caspi, J. Prilusky, and D. Lancet. GeneCards: A novel functional genomics compendium with automated data mining and query reformulation support. *Bioinformatics*, July 1998. available at http://bioinformatics.weizmann.ac.il/cards/CABIOS_paper.html.

[ROL99] M. Tork Roth, F. Ozcan, and L.Haas. Cost models do matter: Providing cost information for diverse data sources in a federated system. *Proc. of VLDB*, 1999.

[RS97] M.T. Roth and P. Schwarz. Don,t scrap it, wrap it! a wrapper architecture for legacy data sources. *Proceedings of the International Conference on Very Large DataBases*, 1997.

[SA99] A. Sahuguet and F. Azavant. Building light-weight wrappers for legacy web data-sources using w4f. In *Proceedings of the VLDB Conference*, 1999.

[SAC$^+$79] P. Selinger, M. Astrahan, D. Chamberlin, R. Lorie, and T. Price. Access path selection in a relational database management system. In *Proceedings of the ACM Sigmod Conf. on the Management of Data*, 1979.

[SMK97] M. Steinbrunn, G. Moerkotte, and A. Kemper. Heuristic and randomized optimization for the join ordering problem. *VLDB Journal*, 6:91–208, 1997.

[TKM99] T. Topaloglou, A. Kosky, and V. Markovitz. Seamless Integration of Biological Applications within a Database Framework. In 7^{th} *International Conference on Intelligent Systems for Molecular Biology (ISBM)*, pages 272–281, Heidelberg, Germany, 1999.

[TRV98] A. Tomasic, L. Raschid, and P. Valduriez. Scaling access to distributed heterogeneous data sources with disco. *IEEE Transactions On Knowledge and Data Engineering*, 1998.

[UFA98] T. Urhan, M. Franklin, and L. Amsaleg. Cost-based query scrambling for initial delays. In *Proccedings of the ACM SIGMOD Conference*, 1998.

[Ull89] J. Ullman. *Principles of Database and Knowledge-Base Systems*, volume II. Computer Science Press, 1989.

[Ull97] J. Ullman. Information integration using logical views. In *Proceedings of the Sixth International Conference on Database Theory*, 1997.

[Uni] UniGene. http://www.ncbi.nlm.nih.gov/unigene/. National Library of Medecine.

[V$^+$01] C. Venter et al. The sequence of the human genome. *Science*, 291:1304–1351, February 2001.

[Vid00] M.E. Vidal. *Mediation Techniques for Multiple Autonomous Distributed Information Sources.* PhD thesis, Universidad Simón Bolívar, Caracas, Venezuela, 2000.

[VP97] V. Vassalos and Y. Papakonstantinou. Describing and using query capabilities of heterogeneos sources. In *Proceedings of the VLDB Conference*, 1997.

[VRG98] M.E. Vidal, L. Raschid, and J-R. Gruser. A meta-wrapper for scaling up to multiple autonomous distributed information sources. In *In the Proceedings of the Intl. Conf. on Cooperative Information Systems*, 1998.

[Wie92] G. Wiederhold. Mediators in the architecture of future information systems. *IEEE Computer*, pages 38–49, March 1992.

[WJ97] et al. White JA. Guidelines for human gene nomenclature. *Genomics*, 45(2):468–471, 1997. See also http://ash.gene.ucl.ac.uk/nomenclature/.

[Won98] L. Wong. Some MEDLINE Queries Powered By Kleisli. In *ACCESS*, June 1998.

[Won00] L. Wong. Kleisli, its Exchange Format, Supporting Tools, and an Application Protein Interaction Extraction. In *IEEE International Symposium on Bio-Informatics and Biomedical Engineering (BIBE)*, Published by IEEE Press, Washington, DC, November 2000.

[YLGMU99] R. Yerneni, C. Li, H. Garcia-Molina, and J. Ullman. Computing capabilities of mediators. In *Proceedings of the ACM SIGMOD Conference*, 1999.

[ZRZB01] V. Zadorozhny, L. Raschid, T. Zhan, and L. Bright. Validating a cost model for wide area applications. In *Proceedings of the Intl. Conf. on Cooperative Information Systems*, 2001.

ior	dependency
$\rightarrow$	functional
$\rightarrow\rightarrow$	multi valued

Figure 3: Dependency of ior

$Input$	binding
$\{ I \}$	$I \in Input$ must be bound
$[I]$	optional binding
$\{ I^+ \}$	set valued binding

Figure 4: Binding of inputs

$Output$	Type and occurrence
O	$O \in Output$ is a value or a hyperlink
O_gid	O is a global identifier
O^+	O is set valued in the output
O^*	O optionally occurs in the output
All	all attribute values (or NULL) returned
$Dup \vert NoDup$	duplicates are (not) eliminated
All^*	all known attribute values returned; output could be semi-structured

Figure 5: Type and occurrence of outputs

PubMed

PubMed(PubMedId_gid, MedlineId_gid, Title, Abstract, Author, GenBankId_gid, SwissProtId_gid ...)
ior_cit1 : { PubMedId$^+$ } → (All^*)
ior_cit2 : { MedlineId$^+$ } → (All^*)
ior_cit3 : { Journal, Date, Volume, Issue, FirstPage, Author } → (All^*)
ior_cit4 : [Journal, Date, Volume, Issue, FirstPage, Author, date ...] →→ (All^*)
ior_cit5 : { Keyword } →→ (All^*)
ior_cit6 : { Phrase } →→ (All^*)

GenBank
Sequence(GenBankId, Locus, Definition, Version, Source, Organism, MedlineId, Sequence, ...)
ior_seq1 : { GenBankId $^+$ } → (Source, MedlineId_url* _ gid, Sequence, ...)
ior_seq2 : [Author*, ModificationDate, Properties, ...] →→ (All^*)
ior_seq3 : { Keyword } →→ (All^*)
ior_seq4 : { Phrase } →→ (All^*)

BLAST
BLAST(Sequence, Database, Score, Query, Subject, Alignment, ...)
ior_blst1: {Sequence, Database} → → { All^* }

GeneCards
GeneCards(HUGO_gid, UniGeneCl_gid, GenBankId_gid, PubMedId_gid, GDBId_gid, ...)
ior_gec1: {HUGO} → { All^* }
ior_gec2: {keyword$^+$} → → { All^* }

UniGene
UniGene(UniGeneCl_gid,LocusLink,OMIM, Homologene, GenBankId_gid, Sequence, ...)
ior_uni1: {UniGeneCl} → { All^* }
ior_uni2: {GenBankId} → { All^* }
ior_uni3: {keyword$^+$} → → { All^* }

SwissProt
SwissProt(SwissProtId_gid, PrimaryAccessionNumber_gid, PubMedId_gid, Sequence, HUGO_gid, GenBankId_gid, ...)
ior_prot1: {SwissProtId} → { All^* }
ior_prot2: {PrimaryAccessionNumber_gid} → { All^* }
ior_prot3: {keyword$^+$} → → { All^* }
ior_prot4: {Author} → → { All^* }
ior_prot5: [Journal, Volume, Year] → → { All^* }

GDB
Gene(GDBId_gid, HUGO_gid, UniGeneCl_gid, SwissProtId_gid, CitationGDBId_id, ...)
ior_gdb1: {GDBId} → (All^*)
ior_gdb2: {HUGO} → (All^*)
ior_gdb3: {UniGeneCl} → (All^*)
ior_gdb4: {Keyword} → → (All^*)
Citation(CitationGDBId, MedLineId_gid, Title, Abstract, Author, ...)
ior_gdb5: {CitationGDBId} → (All^*)
ior_gdb6: {Keyword} → → (All^*)

Figure 6: Examples of Capabilities of Biological Sources

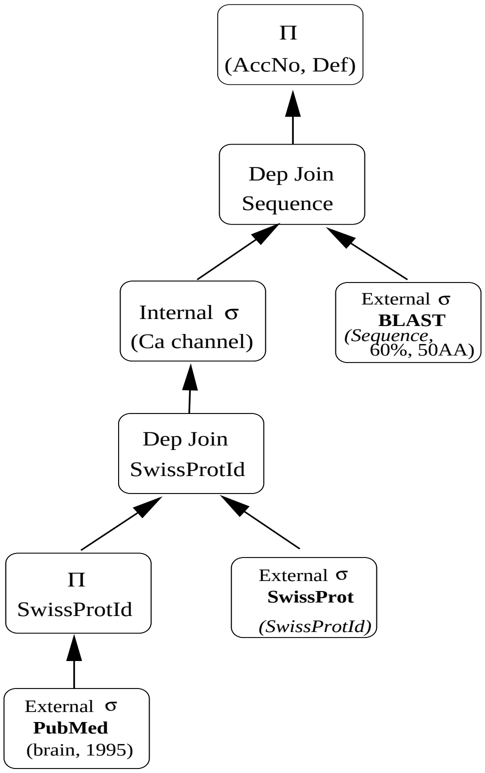

Figure 7: First plan for Query 2

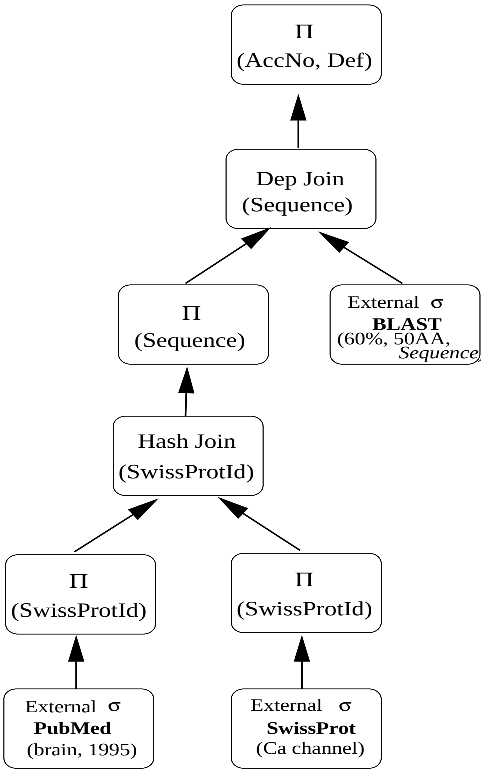

Figure 8: Second plan for Query 2