Integration of Biological Sources: Current Systems and Challenges Ahead

Thomas Hernandez & Subbarao Kambhampati
Department of Computer Science and Engineering
Arizona State University
Tempe, AZ 85287-5406
{th, rao}@asu.edu

Abstract

This paper surveys the area of biological and genomic sources integration, which has recently become a major focus of the data integration research field. The challenges that an integration system for biological sources must face are due to several factors such as the variety and amount of data available, the representational heterogeneity of the data in the different sources, and the autonomy and differing capabilities of the sources.

This survey describes the main integration approaches that have been adopted. They include warehouse integration, mediator-based integration, and navigational integration. Then we look at the four major existing integration systems that have been developed for the biological domain: SRS, BioKleisli, TAMBIS, and DiscoveryLink. After analyzing these systems and mentioning a few others, we identify the pros and cons of the current approaches and systems and discuss what an integration system for biologists ought to be.

1 Introduction

The integration of biological data is just one phase of the entire molecular biology research and genomic hypothesis discovery process. Computer integration – as opposed to human integration – is not actually mandatory in the process: until recently molecular biologists managed to perform their research without any type of non-manual integration. The integration of their repositories and other information sources was then done when researchers needed it, without specifically automating the process, thereby consuming most of a molecular biologist's typical workday. However, now that relevant data is widely distributed over the Internet and made available in different formats, manual integration has become practically infeasible. The amount of data stored in biological databases has indeed grown exponentially over the past decade [15], while simultaneously the number of available biomolecular and genomic sources on the web has increased to more than 500 [11, 39, 2].

Furthermore, the need for effective integration of various bioinformatic sources is also justified by the characteristics of these sources. The main characteristics include

- the highly diverse nature of the data stored,
- the representational heterogeneity of the data,
- the autonomous and web-based character of the sources and the way the data is published and made available to the public,
- the various interfaces and querying capabilities offered by the different sources.

In fact, for these reasons bioinformatic sources are difficult – if not simply time-consuming – for biologists to use in combination with one another.

This paper is a survey that spans the main issues of the integration problematic in the context of biological data sources, and then discusses some of the current approaches and existing systems used to address the challenges raised. The paper is organized as follows. First the characteristics mentioned above are explained in more detail in Section 2. Then Section 3 lists the dimensions along which bioinformatic integration systems can be measured and thus differentiated. Next, Section 4 covers the main integration approaches that have been suggested by the research community. Following the approaches, Section 5 describes some of the major systems that were designed for the integration of biological data and relates each of them to the dimensions and approaches given in the previous sections. Finally, Section 6 contains a few remarks and a discussion of some important open challenges in bioinformatic integration.

2 Characteristics and Challenges

The challenges that must be overcome when integrating heterogeneous bioinformatic sources are numerous and

varied. Listed in this section are the major characteristics of those sources and hence the challenges that must be resolved when trying to integrate them.[1]

Variety of data: The data exported by the available sources cover several biological and genomic research fields. Typical data that can be stored includes gene expressions and sequences, disease characteristics, molecular structures, microarray data, protein interactions, etc. Depending on how large or domain-specific the sources are, they can store different types of data. Furthermore, bioinformatic data can be characterized by many relationships between objects and concepts, which are difficult to identify formally (either because they are abstract concepts or simply because they span across several research topics). Finally, not only can the quantity of data available in a source be quite large, but also the size of each datum or record can itself be extremely large (e.g DNA sequences, 3-D protein structures, etc). This differs from non-scientific (e.g. business data) integration scenarios where there is usually no specific need to address the issue of very large entries.

Representational heterogeneity: The collection of bioinformatic sources has the property that similar data can be contained in several sources but represented in a variety of ways depending on the source. This representational heterogeneity encompasses structural, naming, semantic, and content differences [42]. In other words, not only are they very large but they also each have their own schema complexity (i.e. structural differences). Furthermore, each source may refer to the same semantic concept or field with its own term or identifier, which can lead to a semantic discrepancy between the many sources (i.e. naming and semantic differences). The opposite can also occur, as some sources may use the same term to refer to different semantic objects. Finally, the content differences involve sources that contain different data for the same semantic object, or that simply have some missing data, thus creating some possible inconsistencies between sources. This representational heterogeneity leads to issues such as entity identification across sources and data quality issues, as well as data consistency, redundancy, and curation.

Autonomous and web-based sources: Most of these sources operate autonomously, which means that they are free to modify their design and/or schema, remove some data without any prior "public" notification, or occasionally block access to the source for maintenance or other purposes. Moreover, they may not always be aware of or concerned by other sources referencing them or integration systems accessing them. This instability and unpredictability is further affected by the simple fact that nearly all sources are web-based and are therefore dependent on network traffic and overall availability. An important consequence of the sources being autonomous is that the data is dynamic: new discoveries or experiments will continually modify the source content to reflect the new hypotheses or findings. In fact the only way for an integration system to be certain that it will return the latest data is to actually access the *sources at query time*.

Differing querying capabilities: Individual sources provide their own user-access interface, all of which a user must learn in order to retrieve information that is likely spread across several sources. Additionally the sources often allow for only certain types of queries to be asked, thereby protecting and preventing direct access to their data. These intentional access restrictions force end-users and external systems to adapt and limit their queries to a certain form. Authors in [42] note about the bioinformatic sources that on top of requiring of biologists that they master many different interfaces, some potentially useful information cannot actually be retrieved because of query restrictions and potentially pertinent queries cannot be asked even though the data necessary to answer them is available in the sources.

3 Dimensions of Variation

The existing systems for integrating bioinformatic sources vary along several dimensions, and in this section we will discuss some of the important ones, and the spectrum of variation along those dimensions.

Aim of integration: This refers to the overall goal of the integration system. Some systems are "portal" oriented in that they aim to support an integrated browsing experience for the user (c.f. SRS [30], BioNavigator [5]). Others are more ambitious in that they take user queries and return results of running those queries on the appropriate sources (c.f. TAMBIS [1], DiscoveryLink [26]).

Data model: This refers to the design assumptions made by the integration system as to the syntactic nature of the data being exported by the sources. Some systems have text models of data (c.f. SRS [30], BioNavigator [5]) while others have structured data

[1] Note however that the source properties listed here are not necessarily specific to the biological and genomic domain.

models. In case of structured models, systems differ in terms of the specific model assumed – including relational data, object-relational data (c.f. DiscoveryLink [26]) and nested semi-structured data (e.g. XML and CPL [14]).

Source model: This refers to the assumptions made on the inter-relations between sources. Most systems assume that sources they are integrating are "complementary" in that they export different parts of the schema. Others also consider the possibility that sources may be overlapping (c.f. [4]) in which case *aggregation* of information is required, as opposed to pure integration of information. Integrating complementary sources is often called "horizontal integration" [42] while integrating the overlapping sources is called "vertical integration."

User model: This refers to the type of users that the system is directed towards. The systems that primarily support browsing need to assume very rudimentary expertise on the part of users. In contrast, systems that support user-queries need to assume some level of expertise on the user's part in formulating queries. Some of these systems (c.f. [14]) assume user queries to be formulated in specific languages, while others (c.f. [1]) provide significant interactive support for users in formulating their query.

Level of transparency: This refers to the extent to which the user has control over and/or is required to specify the particular sources that are needed to be used in answering her query. Some systems (e.g. [14]) require the user to select the appropriate sources to be used. Others (e.g. [1]) circumvent this problem by "hard-wiring" specific parts of the integrated schema to specific sources.

4 Integration Approaches

The integration approaches used in the existing systems can be classified first in terms of the data model they use – text, structured data or linked records. For systems that view sources as exporting mainly text, integration involves supporting keyword/text search across the sources. When the sources are viewed as exporting more structured data, there are two broad types of integration approaches, based on whether the data from the sources is "warehoused" or accessed on demand from the sources. Finally, for systems that view sources as exporting linked sets of browsable records, integration involves supporting effective navigation across sources. Since the majority of systems use the (semi-)structured or linked record models, in what follows we will discuss the integration approaches for these in more detail.

4.1 Warehouse integration

Warehouse integration consists in materializing the data from multiple sources into a local warehouse[2] and executing all queries on the data contained in the warehouse rather than in the actual sources. Warehousing emphasizes data translation, as opposed to query translation in mediator-based integration [42]. In fact, warehousing requires that all the data loaded from the sources be converted through data mapping to a standard unique format before it is physically stored locally.

Relying less on the network to access the data obviously helps in eliminating various problems such as network bottlenecks, low response times, and the occasional unavailability of sources. Furthermore, using materialized warehouses allows for an improved efficiency of query optimization as it can be performed locally [19, 12]. Another benefit in the warehouse integration approach is that it allows the system/user to filter, validate, modify, and annotate the data obtained from the sources [13, 23], and this has been noted as a very attractive property for bioinformatics.

This approach however has an important and costly drawback in terms of result reliability and overall system maintenance caused by the possibility of returning outdated results. Warehouse integration must indeed regularly check throughout the underlying sources for new or updated data and then reflect those modifications on the local copy of the data [12].

4.2 Mediator-based integration

Mediator-based integration concentrates on query translation. A mediator in the information integration context is a system that is responsible for reformulating at runtime a query given by a user on a single mediated schema into a query on the local schema of the underlying data sources. Unlike in the warehouse approach, none of the data in a mediator-based integration system is converted to a unique format according to a data translation mapping. Instead a different mapping is required to capture the relationship between the source descriptions and the mediator and thus allow queries on the mediator to be translated to queries on the data sources. Specifying this correspondence is a crucial step in creating a mediator, as it will influence both how difficult the query reformulation is and how easily new sources can be added to or removed from the integration system.

The two main approaches for establishing the mapping between each source schema and the global schema are global-as-view (GAV) and local-as-view (LAV) [19, 29]. In the GAV approach the mediator relations are directly written in terms of the source relations. In other words,

[2]The authors in [23] call it a Unifying Database.

each mediator relation is nothing but a query over the data sources. The GAV approach greatly facilitates query reformulation as it simply becomes a view unfolding process; however handling the addition or removal of a source in a GAV mediator is much more difficult as it requires a modification of the mediator schema to take into account the changes. In a LAV-based mediator every source relation is defined over the relations and the schema of the mediator. It is therefore up to the individual sources to provide a description of their schema in terms of the global schema, making it very simple to add or remove sources but also complicating the query reformulation and processing role of the mediator. Clearly both of these approaches have some positive and negative consequences, but LAV is considered to be much more appropriate for large scale ad-hoc integration because of the low impact changes to the information sources have on the system maintenance, while GAV is preferred when the set of sources being integrated is known and stable.

Several of the bioinformatic integration systems were developed before the advent of the mediated systems, and instead follow the federated database model. A federated database integration system consists of underlying sources which are autonomous components but which also cooperate to allow controlled access to their data. The authors in [41] explain that federated integration can be seen as a middle-ground between no integration (where a user must query each source individually) and total integration (where a user can only query the sources through the integration system). In federated integration the schemas of the component sources are put together to form an integrated schema on which queries will be asked. Seen from this vantage point, mediated systems could be seen as very loosely coupled versions of federated systems.

4.3 Navigational integration

The idea of navigational or link-based integration emerged from the fact that an increasing number of sources on the web require of users that they manually browse through several web pages and data sources in order to obtain the desired information [12]. In fact the major premise and motive justifying this type of integration is that some sources provide the users with pages that would not – or hardly – be accessible without point-and-click navigation. The specific paths essentially constitute workflows in which the output(s) of a source or tool is(are) redirected to the input of the next source until the requested information is reached [9]. In effect, queries are transformed into (possibly several) path expressions that could each answer the query with different levels of satisfaction [34].

Pure navigational integration eliminates relational modeling of the data and instead applies a model where sources are defined as sets of pages with their interconnections and specific entry-points, as well as additional information such as content, path constraints, and optional or mandatory input parameters [10, 29]. The authors in [20] claim that this model effectively allows the representation of cases where the page containing the desired information is only reachable through a particular navigation path across other pages.

Lacroix et al. [28] further explore the path-based approach by analyzing paths between biological sources. Their observation was that multiple physical paths can link two sources. Therefore their goal is to determine properties of links between sources and use these properties to identify the best of several potential execution paths that can answer a given query.

5 Existing Bioinformatic Integration Systems

This section covers a description of some well-known systems that are currently available and tries to characterize each system in terms of the dimensions (c.f. Section 3) and the integration approaches used. The following four systems are discussed in more detail:

- SRS, which is a link-based integration system relying on local index files,
- K2/BioKleisli, which is close to a federated database integration system,
- TAMBIS, which is a mediator-based integration system that uses a global ontology to formulate queries,
- and DiscoveryLink, which is a mediator-based and wrapper-oriented middleware integration system.

After the description of these four systems, other existing bioinformatic integration systems and projects are mentioned. Table 1 provides an overview of the characteristics of each system in terms of the dimensions mentioned in Section 3.

5.1 SRS

The Sequence Retrieval System (SRS) is closer to a keyword-based retrieval system than an integration system. Its approach to bioinformatic integration is to parse flat files or databanks that contain structured text with field names. It then creates and stores an index for each field and uses these local indexes at query-time to retrieve relevant entries [30, 39]. Although extensive indexed entries are kept locally to be used by the query processor at query

	Aim of integration	*Data model*	*Source model*	*User model*	*Level of transparency*	*Overall integration approach*
SRS	Portal, browsing-based	Linked text records	Mostly complementary, some overlap	No critical expertise	Sources specified by user	Navigational, (with local index)
K2/ BioKleisli	Query-oriented	Semi-structured, object-oriented	Mostly complementary	Expertise in query language	Sources specified by user	Mediator-based
TAMBIS	Query-oriented	Structured, object-relational	Mostly complementary	Interactive query formulation	Sources hard-wired by system	Mediator-based
DiscoveryLink	Query-oriented middleware	Structured, object-relational	Mostly complementary, some overlap	Expertise in query language	Sources selected by system	Mediator-based
BACIIS	Query-oriented	Structured, object-relational	Mostly complementary, some overlap	Interactive query formulation	Sources selected by system	Mediator-based
BioNavigator	Portal, browsing-based	Text model	Mostly complementary	No critical expertise	Sources specified by user	Navigational
GUS	Query-oriented	Structured, relational	Mostly complementary	Expertise in query language	N/A[3]	Warehouse
KIND	Query-oriented	Semi-structured, object-oriented	Mostly complementary	Expertise in query language	Sources specified by user	Mediator-based
Entrez	Portal, browsing-based	Linked text records	Mostly complementary, some overlap	No critical expertise	Sources specified by user	Navigational

Table 1: Dimensions of variation (c.f. Section 3) for the existing systems described in Section 5.

time, SRS is not actually a warehouse system as the actual data is neither modified nor stored locally. The other main feature of SRS is that it keeps track of the cross-references between sources. In order to parse the flat files and capture all this information, the system has its own parsing component, ICARUS, which is designed to recognize the presence of links and index all source records using a keyword-based indexing approach. Therefore, while parsing and indexing is performed, the system can identify links that exist between entries in different sources. These links are then used to suggest more results to a user after a query has been processed.

The user query interface is straightforward in SRS [40]. A user first selects which of the many available sources should be queried depending on the type of data expected, and then asks a keyword or gene sequence query on these sources. After the query is processed, the relevant documents in terms of the query keywords are displayed. Additionally, SRS will search in its local index of parsed links for any additional entry that is related in some way to the query. All such links are then made available to the user and grouped by source or by the type of data they point to. In other words, the results of a query in this system are essentially composed of a set of tuples or entries directly retrieved from the initially selected sources, and a set of paths across other sources which lead to information that is related to the query. In fact, a user can browse through a set of sources in a point-and-click type of navigation [12] even after having submitted a very simple query, and find perhaps more relevant or complementary results in the suggested links.

5.2 K2/BioKleisli

BioKleisli is primarily a loosely-coupled federated database system. The mediator on top of the underlying

[3]The GUS warehouse is only composed of one local database so there is no actual transparency issue.

sources relies mainly on a high-level query language that is more expressive than SQL and that provides the ability to query across several sources: the Collection Programming Language, or CPL [8, 14]. CPL requires source specific wrappers to map sub-queries to specific heterogeneous sources, which are accessed through predefined atomic query-functions. The data model used in BioKleisli is an object-oriented type system that is more expressive than the relational model since it includes bags, lists, variants, nested sets and nested records. BioKleisli does not use any global molecular biology schema or ontology that the user could use to formulate queries. This approach therefore requires of the users not only a strong competency in CPL [13] but also a perfect knowledge of the schema and structure of the bioinformatic sources being integrated.

The BioKleisli project is mainly aimed at performing a horizontal integration. In fact, a query attribute is usually bound to an attribute in a single predetermined source; there is essentially no integration of sources with content overlap. Furthermore, no optimization based on source characteristics or source content is performed. The only query optimization that takes place is the reordering of projections, selections, and joins. In fact, the procedural nature of CPL makes the query optimization process difficult.

K2 is the newer version of the BioKleisli system [13]. K2 abandons CPL and replaces it by OQL, a more widely used query language. This change does not modify the overall flow of the system. Queries are still decomposed into subqueries and sent to the underlying sources using data drivers, while the query optimizer remains a rule-based optimizer.

5.3 TAMBIS

TAMBIS (Transparent Access to Multiple Bioinformatics Information Sources) is a mediator-based and ontology-driven integration system [1, 38, 39]. Queries in TAMBIS are formulated through a graphical interface where a user needs to browse through concepts defined in a global schema and select the ones that are of interest for the particular query. The system first expresses the graphical query in GRAIL, a declarative source-independent description logic. The source-independent GRAIL query is then translated into a query internal form (QIF), which is in turn translated into a source-dependent query execution plan in CPL. The choice of CPL was based first on its handling of complex data types well adapted to biological concepts, and second because CPL has a list of available function libraries that are specific to bioinformatic sources and that provide a function-based view of all the underlying sources. Because TAMBIS needs external wrappers, it uses wrappers from the BioKleisli system to access the underlying sources.

The planning and optimization subsystem in TAMBIS only performs reordering of query components; it does not store source statistics or analyze source capabilities. Reordering is based on the cost of individual query components, where the cost combines the predicted time necessary to evaluate a component as well as the expected number of results it will return, i.e. objects or tuples (note that results are not necessarily tuples since not all sources are of relational nature). This optimization therefore does not include any evaluation of sources in terms of content overlap or source availability. In fact, a given concept and its corresponding CPL function are always linked to a predetermined source, which means that even though several sources may contain the same information about a concept, the same source will always be accessed for that particular concept. In short, plans are determined in terms of ordering concepts rather than ordering sources.

It is important to notice that, unlike in many other projects that use ontologies, the ontology defined by TAMBIS is not primarily used for schema mapping between the underlying bioinformatic sources; instead the ontology is a dictionary and classification of biological concepts representing subsumption relationships between concepts. The mapping of ontology concepts to source-dependent CPL functions is done by another subsystem called the Source Model which simply captures which CPL function is related to which ontology concept. Hence the TAMBIS domain ontology mainly serves the purpose of easing the user's task of formulating the query. Overall, the TAMBIS project greatly improves the interface through which a user can pose queries without having to write complex queries nor worry about navigating through sources.

5.4 DiscoveryLink

IBM's DiscoveryLink [25, 26] is a wrapper-oriented bioinformatic integration system. It serves as an intermediary for applications that need to access data from several biological sources. Applications typically connect to DiscoveryLink and submit a query in SQL on the global schema, not necessarily aware of the underlying sources. DiscoveryLink is essentially an integration layer built on the Garlic project technology. It serves as a middleware between the applications and a set of wrappers. The source-specific wrappers must register their data source in order for it to be integrated.

The Garlic technology is mainly a federated database query processor that communicates with source-specific wrappers to determine the optimal plan for a given query and executes the query over possibly several sources [24]. One role for the wrappers is to translate the source data models and describe the data available in the underlying

sources. The data model used by DiscoveryLink is the object-relational model. Additionally the wrappers provide source-specific information about query capabilities that will help the optimizer determine which parts of a query can be submitted to each source.

Using the information provided by the wrappers, the query processor breaks the query into portions that can be handled by different sources. It is then up to the wrapper to produce a plan that the underlying source is capable of executing, and evaluate the execution cost of that plan. The overall cost of all plans is calculated by the optimizer. Several factors are taken into account by the optimizer when computing the cost, such as the local execution cost on each source, network-related costs, selectivity of predicates, and the cost of any remaining operations that cannot be performed by the data sources and that would need to be performed by the DiscoveryLink engine.

After the wrappers have produced their plans and the optimizer decided on the best plan to adopt, the execution engine in will send out individual plans to be executed by the wrappers. The nature and complexity of the plan a wrapper has to execute may vary depending on the query as well as the capabilities of the source. Once the wrappers have performed their plan, the processed data flows from the data sources into the DiscoveryLink engine, which in turn performs any operations that could not be handled by the sources and returns the data to the client application.

Unlike TAMBIS, DiscoveryLink is not a user-end product. A user interface is required to operate on top of DiscoveryLink to elicit queries that are processed and sent to the underlying sources; while in TAMBIS the emphasis is on the source-independent ontology of bioinformatic concepts which enables end-users to formulate their queries easily. Furthermore, TAMBIS and DiscoveryLink also differ in that they do not equally address the optimization of query execution plans in an integration context. In fact TAMBIS concentrates less on query optimization and processing over multiple heterogeneous data sources, which is precisely what DiscoveryLink tries to achieve through the use of its wrappers.

5.5 Other existing systems

Several other existing bioinformatic integration systems or projects are described in this section.

- BACIIS (Biological and Chemical Information Integration System) [33] is an end-user product which was developed following a mediator-based approach combined with extensive use of a knowledge base (KB). The KB contains a domain ontology which serves as a global schema for the system and which captures object classes, attributes, and multiple complex relationships between classes, thereby forming a network structure between classes[4]. The KB also keeps the data source schema which maps the schema of individual sources to the domain ontology. One of the goals of this project is also to derive extraction rules automatically and store them in the source wrappers.

- BioNavigator (part of Biosift's Radia [6]) is a commercially available integration solution which lets users define their preferred execution path for a query. The aim is to allow users to reuse those paths later (similar to macros) for queries of the same type [5, 17]. BioNavigator claims to be capable of integrating heterogeneous sources with a proprietary technology.

- GUS (Genomics Unified Schema) [13, 22] is a system that follows the approach of data warehousing. Maintaining a warehouse allows users of GUS to filter the data and add annotations that a user may want to associate to some retrieved data.

- KIND (Knowledge-based Integration of Neuroscience Data) [21, 31], attempts to combine the use of formal ontologies and conceptual models with source-specific wrappers.

- Entrez [18] is a web-based link-driven federation in which sources are interconnected so that any entry returned from one of the integrated sources will also have related links to the other sources.

- BioHavasu [4] is a project which is much closer to an aggregation system as it concentrates on integrating sources which may potentially all contain the desired information rather than complementary information. The process of selecting between sources is based on the approach used by the online bibliography mediator BibFinder [3], which mines and uses coverage and overlap statistics of the data sources [37].

- Eckman et al. [16] suggest using source capabilities and capability-based rewriting (CBR) to efficiently integrate biological data sources. Source capabilities especially include input/output relationships which exactly describe how a source behaves. Their approach also consists in characterizing in the form of metadata source properties such as access costs and content information.

[4] as opposed to the more common subsumption relationship ontologies which have a tree structure.

6 Discussion

6.1 The ideal system

The first discussion point that should be brought up and that should be addressed by all who plan to build a bioinformatic integration system relates to what the biologists and other researchers actually want as a system. The primary use of such systems is to enable the scientists to acquire some knowledge from large amounts of data, to then formulate hypotheses from the knowledge acquired, and finally perhaps to validate these hypotheses. The amount of work necessary without an integration system is prohibitive, which is why the main goal of these systems should be to automate a maximum number of tasks. Although nothing truly indicates which query interface users prefer (e.g through concept navigation, keyword-based, query-by-example, expressive declarative queries, link navigation, etc.), it is clear that it is up to the system to ensure that users will find what they were looking for in a minimum amount of time and interactions. It is also interesting to note that C. Goble (from the TAMBIS project) admitted that TAMBIS did not actually provide a purely "transparent" access to sources, as it appeared in their experiments that users usually wanted to have the possibility of choosing which sources were accessed and knowing what query plan was to be executed [7]. This tends to show that the system must also be able to provide enough flexibility to the user as well as display the provenance of the data.

Furthermore, because of the span and dynamic nature of biological data, it is also imperative that source representation and source capabilities be automatically extracted. As of today, most source descriptions are obtained through a manual analysis of the source schema or interface by both a domain expert and an integration expert, which usually are two distinct people. Automating the process will reduce the cost and time necessary to develop full-scale integration systems that can keep up with the pace at which biological data is generated.

In addition to the automated extraction of source descriptions, it is important for an integration system to gather source statistics in order to refine the query plans and improve the overall functionality and performance of the system even as the sources evolve (c.f. [36, 37]). Among the essential statistics that should be learned are the coverage of individual sources, the overlap between sources, the average response time, query-dependent response time, and various quality statistics such as freshness of the data and density of each result[5].

Finally, one should note that the integration of scientific data must also take into account the interesting fact that most biologists or researchers value data even though it may be only partially complete and/or potentially incorrect. Any data can indeed be relevant to a scientific researcher. Thus we cannot wily nilly delete or ignore incomplete datasources.

6.2 Discussion on integration approaches

As Section 4.1 pointed out, the warehouse approach can provide two clear advantages. First, it simplifies query optimization and processing by storing the data locally according to a single global schema. Second, it enables users to add their own annotations to some stored data and specify some filtering conditions to clean the data as it is stored locally. Although this would indeed be a definitive improvement for the user of the system, it is still unclear how this process could be achieved efficiently, and more specifically how the data could effectively be validated or modified without requiring costly and time-consuming human intervention, as well as extensive domain expertise. The data retrieved and integrated in the warehouse will indeed eventually have to be converted into a warehouse-specific format. Furthermore, as mentioned earlier, data warehousing in general must still face the vast problem of handling updates in the data sources, which here would be an even greater challenge as the data contained in the warehouse may be modified and annotated, and therefore always different from the data in the underlying sources. It is however interesting to note that despite those negative aspects, authors in [23] have recently suggested that a large curated warehouse was the only effective approach, given the pervasive data inconsistency across sources.

Although Section 4.2 explained the GAV and LAV approaches for mediator-based integration, it is interesting to note that none are truly implemented in the biological integration systems mentioned in this survey. A reason may simply be that biologists started working on biological databases and repositories long before web-based data integration became a research field in Computer Science. In fact, most systems used initially were either warehouses or federated databases, which have survived and evolved since. Only relatively recently has the field lead to wrapper-oriented or navigational approaches.

The concept of navigational integration has in fact not yet established itself as a true alternative to the other, more common integration approaches. In fact, the authors in [12] even doubt that link-driven integration is pertinent to bioinformatic integration because it does not offer enough querying functionality and secondly because it is not well adapted to the extent of the data available and to the ever-changing sources. However, one cannot overlook this type of integration, as path-based optimization of queries seems like a promising new direction.

[5] *density* refers to the degree of completeness of each of the answer records returned by a source (c.f. [35]).

Much like TAMBIS and K2, most of the currently widely used integration systems only address the horizontal dimension of data integration. In integrating only sources that have complementary data, an integration system does not take into account the potential overlapping aspect of sources or the probable incompleteness of some sources. Restricting the integration process to simply combining data from sources that contain different types of information for the same semantic entity limits the capability of a system, especially in terms of reliability and completeness. Moreover, it prevents one from optimizing the system to:

- deal with absence of data or contradictory data,
- identify bad-quality data,
- use response times to improve query execution,
- and select sources that have a higher chance of returning better or more results.

A purely horizontal integration system cannot address these issues of effectiveness and efficiency. In fact, aggregation of information and sources is also necessary. Considering the possibility of having several candidate sources for the same mediator relation would essentially allow a system to address these issues. DiscoveryLink makes an attempt to solve the problem of selecting between several potential sources by using the estimated query processing cost given by the individual wrappers, although the overlap and coverage point of view of optimization and source selection is not considered.

7 Conclusion

This survey justified the need for systems that provide an integration of bioinformatic sources as there exists a real demand from biological researchers who are now overwhelmed by the amount of work necessary to manually go through the integration process. Dimensions of variation between systems and general integration approaches were then described. Systems that try to solve the challenges posed by such a large-scale and domain-specific integration were then exposed. The current approaches can be divided into several classes, including warehouse, mediator, and navigational-based integration systems. However, after a description of the major systems used by biologists today (SRS, K2/BioKleisli, TAMBIS, and DiscoveryLink), it was clear that none of them follow the guidelines of exactly one approach. Often aspects of different approaches are in fact joined together to build these integration systems.

A discussion followed by pointing out that the ideal integration system should truly take into consideration the wishes of those who will be using the system. It was mentioned that reducing the time and interactions required to use the system were the key issues. Automating a maximum number of tasks is the ultimate goal systems should aim at in order to integrate data at the same pace as the field is moving. A discussion on current approaches was then presented. The lack of aggregation systems, which integrate sources containing semantically similar data, also known as vertical integration, was pointed out. By restricting themselves to the integration of sources that contain complementary data, or horizontal integration, current systems cannot compensate for the possible absence of data in a source, contradictory data, low response times and other important factors that would improve query optimization and execution time.

Acknowledgments

This work is supported by the ASU ET-I^3 initiative grant ECR A601. We wish to thank Louiqa Raschid for her comments on the initial version of this survey. We also thank all participants of the ET-I^3 initiative at Arizona State University as well as researchers at TGen who attended discussions related to this work.

References

[1] P. Baker, A. Brass, S. Bechhofer, C. Goble, N. Paton, and R. Stevens. TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources. *In Proceedings of the Sixth International Conference on Intelligent Systems for Molecular Biology (ISMB98)*, 1998.

[2] A. D. Baxevanis. The Molecular Biology Database Collection: 2003 Update. *Nucleic Acids Research*. 31(1), 1-12, 2003.

[3] BibFinder. http://kilimanjaro.eas.asu.edu, 2004.

[4] BioHavasu. http://phanxipan.eas.asu.edu, 2003.

[5] BioNavigator Solutions. http://www.bionavigator.com, 2003.

[6] BioSift Radia. http://www.biosift.com/products/radia/-radia.asp, 2003.

[7] O. Boucelma, S. Castano. C. Goble, V. Josifovski, Z. Lacroix and B. Ludascher. Report on the EDBT'02 Panel on Scientific Data Integration. *ACM SIGMOD Record*, 31(4), 2002.

[8] P. Buneman, S. Davidson, K. Hart, C. Overton, L. Wong. A Data Transformation System for Biological Data Sources. *In Proceedings of the 21st VLDB Conference*. 1995.

[9] David Buttler, Matthew Coleman1, Terence Critchlow1, Renato Fileto, Wei Han, Ling Liu, Calton Pu, Daniel Rocco, Li Xiong. Querying Multiple Bioinformatics Data Sources: Can Semantic Web Research Help? *ACM SIGMOD Record*, 31(4), 2002.

[10] A. Cali, D. Calvanese, G. De Giacomo and M. Lenzerini. On the Expressive Power of Data Integration Systems. *In Proceedings of the 21st International Conference on Conceptual Modeling (ER 2002)*. 2002.

[11] DBCAT, The Public Catalog of Databases. Infobiogen. http://www.infobiogen.fr/services/dbcat, 2003.

[12] S. Davidson, C. Overton and P. Buneman. Challenges in Integrating Biological Data Sources. *Journal of Computational Biology*. Vol 2, No 4, 1995.

[13] S. Davidson, J. Crabtree, 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.

[14] S. Davidson and L. Wong. The Kleisli Approach to Data Transformation and Integration. 2001.

[15] EMBL Nucleotide Sequence Database Statistics, http://-www3.ebi.ac.uk/Services/DBStats, 2003.

[16] B. Eckman, Z. Lacroix, L. Raschid. Optimized Seamless Integration of Biomolecular Data. *Proceedings of the IEEE 2nd International Symposium on Bioinformatics and Bioengineering Conference*, 2001.

[17] Entigen. BioNavigator - BioNode & BioNodeSA: Overview. http://www.entigen.com/library, 2001.

[18] Entrez - Search and Retrieval System. http://www.ncbi-.nlm.nih.gov/Entrez, 2003.

[19] D. Florescu, A.Y. Levy, and A.O. Mendelzon. Database Techniques for the World-Wide Web: A Survey. *ACM SIGMOD Record*, 27(3), 59-74, 1998.

[20] M. Friedman, A. Levy, and T. Millstein. Navigational Plans For Data Integration. *In Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 67-73, 1999.

[21] A. Gupta, B. Ludascher, and M. E. Martone. Knowledge-Based Integration of Neuroscience Data Sources. *In Intl. Conference on Scientific and Statistical Database Management (SSDBM)*. 2000.

[22] The GUS Platform for Functional Genomics. http://-www.gusdb.org, 2003.

[23] J. Hammer and M. Schneider. Genomics Algebra: A New, Integrating Data Model, Language, and Tool Processing and Querying Genomic Information. *In Proceedings of the 2003 CIDR Conference*. 2003.

[24] L. M. Haas, R. J. Miller, B. Niswonger, M. Tork Roth, P. M. Schwarz, and E. L. Wimmers. Transforming Heterogeneous Data with Database Middleware: Beyond Integration. *IEEE Data Engineering Bulletin*, 22(1), 31-36, 1999.

[25] L. Haas, P. Kodali, J.E. Rice, P. Schwarz, W.C. Swope. Integrating Life Sciences Data - With a Little Garlic. *IEEE International Symposium on Bio-Informatics and Biomedical Engineering (BIBE'00)*, 2000.

[26] L. Haas, P. Schwarz, P. Kodali, E. Kotlar, J. Rice and W. Swope. DiscoveryLink: A System for Integrated Access to Life Sciences Data Sources. *IBM Systems Journal*, 40(2), 489-511, 2001.

[27] C. Knoblock, S. Minton, J. L. Ambite, N. Ashish, I. Muslea, A. G. Philpot, and S. Tejada. The ARIADNE Approach to Web-based Information Integration. *The Journal on Intelligent Cooperative Information Systems (IJCIS)* Special Issue on Intelligent Information Agents: Theory and Applications, 10(1/2), 145-169, 2001.

[28] Z. Lacroix, F. Naumann, L. Raschid, and M-E. Vidal. Exploring Life Sciences Data Sources. *Proceedings of IJCAI-03 Workshop on Information Integration on the Web*, 2003.

[29] M. Lenzerini. Data Integration: A Theoretical Perspective. *ACM Symposium on Principles of Database Systems (PODS)*, 2002.

[30] R. Lopez. SRS - Sequence Retrieval System. Presentation. http://www.pdg.cnb.uam.es/cursos/BioInfo2001/pages/-SRS/. Universidad Autonoma de Madrid, 2001.

[31] B. Ludascher, A. Gupta, M. E. Martone. Model-Based Mediation with Domain Maps. *17th Intl. Conference on Data Engineering (ICDE)*, 2001.

[32] I. Manolescu, D. Florescu, and D. Kossmann. Answering XML Queries over Heterogeneous Data Sources. *In Proceedings of the 27th VLDB Conference*. 2001.

[33] Z. Ben Miled, N. Li, M. Baumgartner and Y. Liu. A Decentralized Approach to the Integration of Life Science Web Databases. *Informatica*. 27(1), 2003.

[34] P. Mork, A. Halevy, and P. Tarczy-Hornoch. A Model for Data Integration Systems of Biomedical Data Applied to Online Genetic Databases. *In Proceedings of the Symposium of the American Medical Informatics Association*, 2001.

[35] F. Naumann and J. C. Freytag. Completeness of Information Sources. Technical Report HUB-IB-135, Humboldt University of Berlin, 2000.

[36] Z. Nie and S. Kambhampati. Joint Optimization of Cost and Coverage of Query Plans in Data Integration. *In Proceedings of the 10th Intl. Conference on Information and Knowledge Management (CIKM)*, 2001.

[37] Z. Nie and S. Kambhampati. A Frequency-based Approach for Mining Coverage Statistics in Data Integration. *20th Intl. Conference on Data Engineering (ICDE)*, 2004.

[38] N. Paton, R. Stevens, P. Baker, C. Goble, S. Bechhofer, and A. Brass. Query Processing in the TAMBIS Bioinformatics Source Integration System. *In Proc. SSDBM*, 138-147. IEEE Press, 1999.

[39] N. Paton and C. Goble. Information Management for Genome Level Bioinformatics. *VLDB 2001 Tutorial*. 2001.

[40] SRS Documentation. SRS at the European Bioinformatics Institute. http://srs.ebi.ac.uk/, 2003.

[41] A.P. Sheth and J.A. Larson. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Computing Surveys*. 22(3), 183-236, 1990.

[42] W. Sujansky. Heterogeneous Database Integration in Biomedecine. *Methodological Review, Journal of Biomedical Informatics*, 34, 285-298, 2001.