BIOINFORMATICS ORIGINAL PAPER

Vol. 21 no. 4 2005, pages 430–438
doi:10.1093/bioinformatics/bti187

An architecture for biological information extraction and representation

Aditya Vailaya, Peter Bluvas, Robert Kincaid, Allan Kuchinsky, Michael Creech and Annette Adler*

Agilent Laboratories, 3500 Deer Creek Road, MS 26U-16, Palo Alto, CA 94304, USA

Received on May 21, 2004; accepted on November 11, 2004
Advance Access publication December 17, 2004

ABSTRACT

Motivations: Technological advances in biomedical research are generating a plethora of heterogeneous data at a high rate. There is a critical need for extraction, integration and management tools for information discovery and synthesis from these heterogeneous data.

Results: In this paper, we present a general architecture, called ALFA, for information extraction and representation from diverse biological data. The ALFA architecture consists of: (i) a networked, hierarchical, hyper-graph object model for representing information from heterogeneous data sources in a standardized, structured format; and (ii) a suite of integrated, interactive software tools for information extraction and representation from diverse biological data sources. As part of our research efforts to explore this space, we have currently prototyped the ALFA object model and a set of interactive software tools for searching, filtering, and extracting information from scientific text. In particular, we describe BioFerret, a meta-search tool for searching and filtering relevant information from the web, and ALFA Text Viewer, an interactive tool for user-guided extraction, disambiguation, and representation of information from scientific text. We further demonstrate the potential of our tools in integrating the extracted information with experimental data and diagrammatic biological models via the common underlying ALFA representation.

Contact: aditya_vailaya@agilent.com

1 INTRODUCTION

The completion of the draft sequence of the human genome has heralded a new era in molecular biology by opening avenues for measurement of a large number of genes or proteins in a single experiment. This has led to a huge paradigm shift for life scientists, from the traditional research limited to experimenting with small, specific, and approximate biological models to the newer paradigm of high throughput experiments. These experiments have the potential to revolutionize academic and industrial research and discovery, with breakthroughs in areas such as identifying genetic causes of disease, predicting an individual's response to drug treatment, identifying biological drug targets, and deepening the basic understanding of evolution and workings of biological organisms. However, there still exist tremendous problems in data and information integration and management before the potential of these technologies can be realized.

The new experimental paradigm has led to an emergent data-centric approach to research in molecular biology. The emphasis is on mining the large amounts of data generated by these high throughput experimental studies to yield specific targets for further research. This approach is complementary to the traditional hypothesis-centered research, where biologists approached their research problems with specific hypotheses, and experiments were conducted to refute or support these. In the traditional hypothesis-centered paradigm, biologists worked within the realm of their domain expertise, where they had sufficient knowledge to develop and test various hypotheses. However, the new data-centric approach is forcing biologists to move out of narrowly focused areas to work more broadly with large numbers of genes or proteins that they are otherwise unfamiliar with, i.e., they are forced to incorporate broader range of scientific knowledge drawn from disciplines beyond their own. A biologist now has the need to draw insights across multiple data (e.g., comparing proteomic vs. gene expression information) and data types (e.g., comparing information from scientific text, pathway diagrams, and experimental data). Clearly, no one central repository (either technological or human) exists with all the biological knowledge. Thus, the challenge lies in providing biologists with easier access to these highly fragmented and distributed sources of data and information, and a means of viewing, navigating, and synthesizing this information.

One major source of knowledge is scientific literature. For generations the research community has represented and shared its knowledge in the form of publications.

*To whom correspondence should be addressed.

Vailaya *et al*. (2004) An architecture for biological information extraction and representation. Symposium on Applied Computing, Proceedings of the 2004 ACM symposium on Applied computing, 103–110; http://doi.acm.org/10.1145/967900.967924

More recently these are being shared freely in a digital and computationally accessible format (mostly in the form of abstracts, such as the PubMed repository at NCBI (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=PubMed) and possibly as full articles in the future). With the explosion in molecular biology literature (for example, there are over 14 million abstracts in PubMed as of March 2004), searching for relevant information is becoming extremely difficult with the user left to manually sift through large number of retrieved documents. Therefore, simple access to digital forms of text isn't sufficient for analyzing high throughput experimental data and a number of research groups have started to address the problem of automatic information extraction (IE) from text (Collier *et al*., 2000; Friedman *et al*., 2001; Fukuda *et al*., 1998; Humphreys *et al*., 2000; Iliopoulos *et al*., 2001; Krauthammer *et al*., 2000; Ng and Wong, 1999; Palakal *et al*., 2002a,b; Park *et al*., 2001; Rindflesch *et al*., 2000; Sekimizu *et al*., 1998; Stephens *et al*., 2001; Wong, 2001; Yakushiji *et al*., 2001). However, there are several limitations of automatic text extraction. It behaves as a black box to an end user not knowledgeable in the text extraction domain, its accuracy and efficiency are not easily computable, and it often fails to capture a user's true context (in that its behavior is uniform across all users). Moreover, automated extraction systems are very specific in terms of the problem they are solving. A study of biologists in pharmaceutical industry and academic research labs indicates that most biologists do not completely trust automated text extraction systems, for the above-mentioned reasons (O'Day *et al*., 2001).

Manual extraction of information by a user, on the other hand, is true to the user's intent, the user trusts it, and in this method the user has a better understanding of the accuracy of the extracted information. Also, capturing the user's context is implicit with manual extraction. However, manual extraction is tedious and time consuming, and increases in difficulty as data and models grow in complexity. Moreover, it becomes potentially impossible to manually extract information in a high throughput manner. Therefore, in order to overcome the limitations and utilize the advantages of the automated and manual approaches, we propose an interactive, user-guided approach to information search and extraction. The following verbal quote from a biologist in the Text Mining panel discussion at the Pacific Symposium on Biocomputing 2002 (PSB 2002), at Kauai, Hawaii, succinctly captures the motivation behind our approach. '*Biologists want interactive tools to help them do their job better and not fully automated tools whose reliability is hard to judge. Thus text mining should not be an end product, but an interacting tool allowing users to extract information they are interested in.*'

Finally, extraction of information from text is not an end in itself. Biologists need to relate the extracted information with other sources of information, such as experimental data or diagrammatic biological models. In other words, tools are needed for (i) robust and reliable extraction and representation of information from text, ideally in a high throughput manner; (ii) providing biologists access to the relevant and interesting information based upon the context of their experiments or research; (iii) incorporating the relevant information in experimental data analysis and model generation; and (iv) linking the relevant information, experimental data, and models for reuse, collaborative sharing, and knowledge synthesis. In this manuscript, we describe our efforts to address the above issues via a suite of integrated, interactive, user-guided software tools. Our approach utilizes some of the more reliable information retrieval and extraction technologies to automatically *query* multiple information repositories, *filter* the retrieved results based on the user's interests, and *identify* relevant information (what we refer to as potential entities (or concepts) and their interactions (or relations)) in the filtered text corpus. However, rather than completely automate the extraction process, we provide the users interactive tools to *guide* the extraction process. We have further developed an architecture to *represent* the extracted information in a hierarchical hyper-graph data structure. This architecture, referred to as ALFA, allows for easy transformations between textual, experimental, and diagrammatic biological data by providing means for a standardized and structured representation of information present in these data sources.

The rest of the manuscript is organized as follows. Section 2 presents a brief review of the information extraction literature. In Section 3, we discuss our architecture for representing information from text, diagrammatic biological models and experimental data. We describe our tools for searching and extracting information from scientific text in Section 4. We present applications of our technology in Section 5 and finally conclude in Section 6.

2 IE REVIEW

Information extraction (IE) from text is defined as taking free form text and producing a structured representation. Common methods for extracting information from text vary from simple statistical methods such as term co-occurrences (Iliopoulos *et al*., 2001; Stephens *et al*., 2001) or Hidden Markov Models (HMMs) for term identification and classification (Collier *et al*., 2000) to computationally intensive structural methods such as Natural Language Processing (NLP) techniques, which may utilize rule-based grammars, part of speech taggers and parsers (Friedman *et al*., 2001; Fukuda *et al*., 1998; Humphreys *et al*., 2000; Krauthammer *et al*., 2000; Ng and Wong, 1999; Palakal *et al*., 2002a,b; Park *et al*., 2001; Rindflesch *et al*., 2000; Sekimizu *et al*., 1998; Wong, 2001; Yakushiji *et al*., 2001). Irrespective of the method (statistical or structural) the common tasks in information extraction from text can be broadly classified as follows (Humphreys *et al*., 2000):

(i) Named Entity Recognition and Template Element Filling: Identifying names of genes, proteins, drugs, etc., in text and

classifying the named entities into classes of interest such as genes, proteins, organelles, cells, organs, diseases, drugs, etc. (proper noun identification and classification) (Collier *et al*., 2000; Fukuda *et al*., 1998; Krauthammer *et al*., 2000; Palakal *et al*., 2002b). Automatic recognition of gene or protein names is a very hard problem due to a number of factors, such as a lack of proper naming conventions, ambiguous lexical cues and constraints (multiple representations such as TNFA, TNF-alpha, tnf alpha, TNFalpha; aliases like JNK, MAPK8, JNK1, PRKM8, SAPK1, JNK1A2, JNK21B1/2; or ambiguous acronyms like ALT for alanine aminotransferase), long compound word names (tumor necrosis factor alpha), and ambiguous symbols (mAB, which can stand for 'monoclonal antibody' or 'male abnormal' gene).

(ii) Template Relation Filling: Extracting relations between named entities such as protein-protein interactions, protein localization (protein-cell interactions), protein-disease interactions, disease-drug interactions, etc. (Friedman *et al*., 2001; Fukuda *et al*., 1998; Humphreys *et al*., 2000; Ng and Wong, 1999; Park *et al*., 2001; Rindflesch *et al*., 2000; Sekimizu *et al*., 1998; Wong, 2001; Yakushiji *et al*., 2001). Automatic extraction of relations becomes very hard and error prone due to problems in disambiguating subject from object and handling common structures in natural language, such as coordination (matching nouns to verbs), apposition (when to generalize), and anaphora (handling pronouns). A thorough analysis of the problems faced in automatic recognition of biological terms and extraction of relations among them can be found in (Park *et al*., 2001; Wong, 2001).

(iii) Discourse Analysis: Determine context around the extracted entities and relations, such as organism, tissue, cell type, disease model, control vs. treated, localization—membrane, cytoplasm, nucleus, etc. (Iliopoulos *et al*., 2001). Discourse analysis is an even harder problem than extraction of interactions, since this requires a more comprehensive language analysis. The context of an interaction may be widely separated from the text of the interaction, thus compounding the problems due to anaphora resolution, coordination, and apposition.

Recently Palakal *et al*. (2002a) have proposed an intelligent information management system to sift through vast volumes of heterogeneous data. Their tool, BioSifter, automatically retrieves relevant text documents from biological literature based on a user's interest profile. The tool acts as a filter by significantly reducing the size of the information space. The filtered data obtained through BioSifter is relevant as well as much smaller in dimension compared to all the retrieved data. The expectation is that BioSifter would significantly reduce the complexity associated with the next steps in information management, that of transformation of information to knowledge.

As in the system defined in Palakal *et al*. 2002a, we are developing integrated and interactive software tools to identify, filter and extract relevant information from heterogeneous data sources. We utilize some of the more reliable information retrieval and extraction technologies to automatically search and identify relevant information from text. We further provide an architecture for representing the extracted information in a common structured format, which allows for easy transformations between textual, experimental, and diagrammatic biological data.

3 INFORMATION REPRESENTATION

We have developed an architecture, referred to as ALFA, for qualitative representation of biological information. The goals of ALFA are to capture and represent, in a structured manner, information from free form text, experimental data, and diagrammatic models by defining a common hierarchical hyper-graph data structure to represent the underlying information. While the emphasis of ALFA is to provide a means for representing information present in various sources of biological data in an abstract yet structured manner, the hierarchical hyper-graph data structure also provides a visual and computational framework to query, modify, and visualize the represented information. Moreover, this model lends itself to extensions for developing quantitative models of biological processes.

3.1 ALFA architecture

The ALFA architecture consists of an object model and an API layer to access the underlying objects. The architecture provides mechanisms to annotate objects, add a list of properties to the objects, and attach proprietary and external ontologies as classifiable properties of the objects. The architecture further incorporates a set of *native* (scientific text, experimental data, and diagrammatic biological models) data viewers, which aid in extraction of information from each of these native data sources into the ALFA object model and conversely to incorporate information stored as ALFA objects into the respective viewers.

Figure 1 shows a block diagram of the ALFA architecture. Each of the native data viewers, ATV for ALFA Text Viewer, ANV for ALFA Network Viewer, and AEV for ALFA Experimental data Viewer, is an interactive tool that aids the user to guide information extraction from the respective native data source. Likewise, each tool can be used as a means of viewing its respective data type. As shown in the figure, the data viewers incorporate *user context* for identifying information of interest to a user. We define user context as a set of ALFA objects that are of interest to a user. They can be a list of genes or proteins, terms identifying experimental processes, specific interactions of interest, etc. Further details of the use of user context for identifying interesting information from text is described in Section 4.2 and its use in user-guided information extraction (in terms of ALFA objects) is described in Section 4.3. The ALFA architecture also consists of a base ontology (a hierarchical classification scheme for common

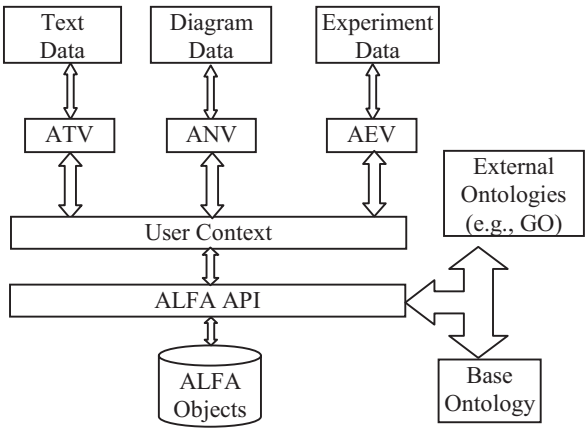

Fig. 1. A block diagram of ALFA architecture.

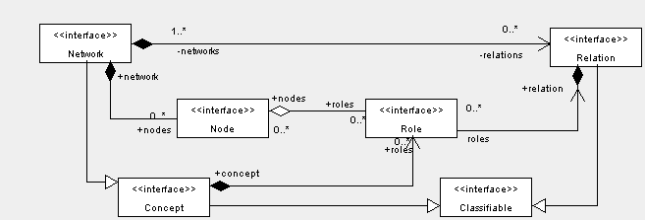

Fig. 2. A simplified UML Diagram of ALFA Object Model.

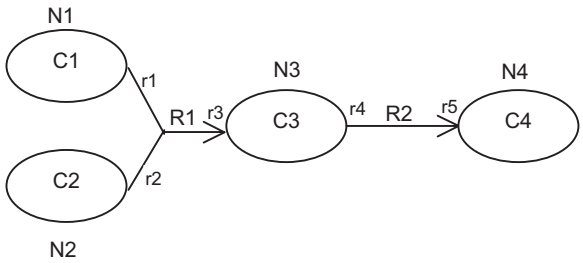

Fig. 3. An example of ALFA objects representing information in the sentence, 'HIP-55 binds to HPK1 and regulates JNK1 signaling cascade'; C1-C4 represent the concepts HIP-55, HPK1, the bound molecule of HIP-55 and HPK1, and JNK1 signaling cascade (a concept that is also a network representing the JNK1 signaling cascade, but not represented here), respectively. R1 and R2 represent the 'binding' and 'regulates' relations, respectively; r1-r5 represent the upstream and downstream roles played by the concepts C1-C4; N1-N4 represent nodes in the representation.

molecular biology terms, such as genes, proteins, metabolites, diseases, processes, cells, tissues, organisms, etc.), which can be assigned to ALFA objects as properties (details of the base ontology are not included due to space limitations). Further, ALFA provides a mechanism for easy incorporation of external ontologies, such as the GO classification scheme (Gene Ontology, 2004, http://www.geneontology.org/). These ontologies can be used to classify ALFA objects, which can be used to query, sort, or filter ALFA objects. Since an ontology may have inherent relations between the ontological terms, these relationships can also be used for defining rules and performing automatic rule checking during interactive information extraction.

3.2 ALFA object model

Figure 2 shows a simplified UML diagram of the hierarchical ALFA object model, which consists of the following objects: *concept*, *relation*, *role*, *node*, *network*, and *classifiable*. For sake of clarity, Figure 3 displays an example of an excerpt of text and its respective ALFA representation.

A *concept* refers to a biological entity, such as a gene, protein, molecule, ligand, disease, drug or other compound, process, etc. A list of properties can be attached to every concept, which may include name, aliases, sequence information, contextual information about the concept, such as state (active, inactive, post-translational modifications, etc.), location, etc. A *relation* is an interaction between multiple concepts. Each concept plays a specific *role* in the relation. Currently defined roles in ALFA include upstream, downstream, mediator, container, and unknown. As with a concept, a list of properties can be attached to a relation to specify its name, type (such as activation, inhibition, catalytic, etc.), location, etc. A *node* object connects multiple relations together by connecting the roles of a common concept between different relations. For example, the node N3 in Figure 3 connects the roles r3 and r4 of the concept C3 in two different relations. If the two roles of concept C3 were not connected, then two different node objects would be created for the two roles of C3. A node can thus act as a bridge between two or more relations. A *network* consists of a list of relations and nodes. Hierarchical structure is incorporated into ALFA via networks. A network is also a concept and when represented as such, abstracts its list of relations to the user. For example, the following relation, 'epinephrine inhibits glycolysis' would represent epinephrine as an upstream concept and glycolysis as a downstream concept of an inhibitory relation. However, the process of glycolysis can also be represented as a set of relations, specifying the steps in the anaerobic breakdown of glucose to pyruvate yielding two molecules of ATP, and stored as a network. Therefore, we can hierarchically represent biological processes by allowing a network to be a concept. A *classifiable* object defines an ontological term. Both concept and relation objects are also classifiable objects to which ontological terms can be attached.

4 INTERACTIVE TEXT EXTRACTION

As part of the ALFA architecture, we have developed interactive software tools for searching, filtering, and extracting information from scientific text. The process for interactive text extraction involves two sub-processes: (i) a meta-search

engine called BioFerret, for searching and filtering information, and (ii) an interactive tool called ALFA Text Viewer (ATV) for user-guided information extraction. BioFerret allows the user to search public and proprietary databases based on keywords. The retrieved results are then automatically processed to identify interesting and relevant documents based on the user's context terms. The filtered set of relevant documents is then processed by ATV, which aids in the conversion of interesting information present in the text document to ALFA objects. The extracted ALFA objects can then be visualized as an ALFA network diagram, compared with preexisting ALFA objects, which may have been extracted from biological diagrams, experimental data, or other text, etc. Section 5 describes in detail further applications of the ALFA architecture in integrating heterogeneous data. We next describe the text extraction process in further detail.

4.1 User-guided meta-search

BioFerrert is a meta-search tool for automatically querying multiple search engines (both public and proprietary) in order to aid biologists facing the daunting task of manually searching for information from these multiple sources. Figure 4 shows a screen shot of BioFerret. When a query is entered, it is submitted to multiple user-selected search engines, the retrieved results are fetched from their respective sources and merged to remove duplicates, the resultant corpus is clustered based on hierarchical clustering (Jain and Dubes, 1998) using a bag of words feature vector, and the documents are classified into a set of user-defined categories using a probabilistic machine learning approach (details are beyond the scope of this manuscript). The documents are further ranked according to the weighted occurrence of the search terms in the document's title and text. These data mining techniques are provided for organizing and navigating the retrieved documents.

BioFerret can run queries in a batch mode for automatically dispersing multiple queries to the multiple search engines and databases. The results are merged into a single set of results and organized via ranking, clustering and categorization. Queries can also be run in an alias resolution mode, wherein every query term is automatically expanded to an OR of all its aliases. Section 4.3.1 describes the alias resolution method in more detail. BioFerret also provides a mechanism to store search results in a Microsoft Access™ database, for re-use, sharing, or re-analysis using different context terms. BioFerret currently provides access to public databases such as PubMed, OMIM, USPTO, and some common web search engines, such as HotBot and AltaVista™.

4.2 User context and relevance ranking

We have developed a simple technique based on user defined lexicons called "user context" to automatically sift through the potentially large corpus of search results to quickly identify information of interest to the user. A user context is defined as

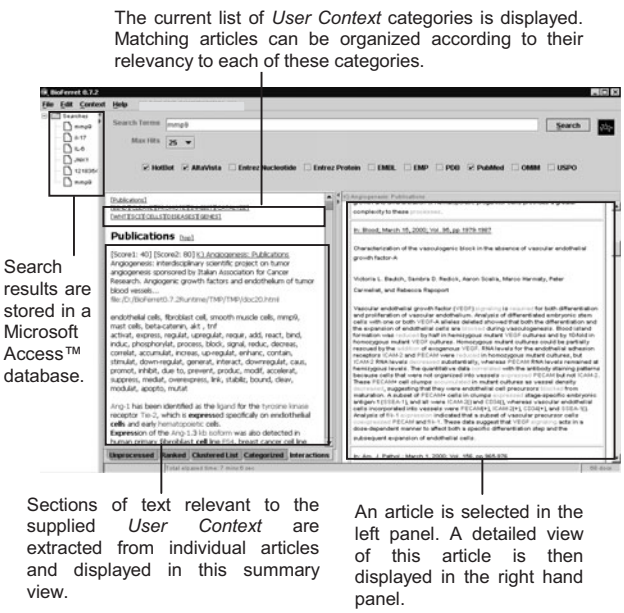

Fig. 4. A screenshot of BioFerret.

a set of ALFA objects the user is interested in. For example, it can be a list of genes or proteins in a pathway, a list of experimental procedures, a list of disease conditions, a list of drugs, a list of specific interactions of interest, etc. The user context can vary for different users and is incorporated at run-time in BioFerret to identify sentences in a document that match the user context. Matching is performed in terms of simple absence/presence of context terms in the given sentence. Users can also set multiple contexts to process a corpus of documents. A bucket is maintained for each of the user contexts and when a sentence in a document contains a user context term (word or phrase), the sentence and the document are added to the respective bucket. The matching sentence is referred to as an 'interesting' sentence. Each of the documents in the corpus (search result) is assigned a score based on the number of matching context terms and the number of 'interesting' sentences. The documents are then ranked according to their scores and organized into multiple categories, one for each of the user contexts.

The set of 'interesting' sentences is further processed to identify potential concepts/entities (gene and protein symbols, drugs, compounds, processes, etc.) and their potential relations/interactions. We have used the user context files and an open source tool, BNS (Kincaid *et al*., 2002, http://openbns.sourceforge.net/), for identifying gene and protein names in text. BNS is an LDAP-based system that uses the Lightweight Directory Access Protocol (LDAP) (Wahl *et al*., 1997) to provide high performance and scalable access to data derived from an identifier-mapping database (LocusLink in this case). Every sentence is tokenized into words and stemmed using the Porter stemmer (Porter, 1980).

The stemmed words are filtered using a dictionary of common English words, which is based on the dictionary provided with the UNIX operating system. BNS is used to query for words that are either capitalized in text or not contained in the dictionary. Words found in a BNS lookup are marked as potential gene or protein symbols. The user context files are also used to identify other potential entities of interest, which may have been missed by the BNS lookup, to the user, such as drugs, compounds, processes, etc. All the words in the text document that match a user context term are also marked as potential entities of interest.

An interactions specific user context, referred to as the interaction lexicon, is also provided in BioFerret for identifying potential interactions between the identified genes or proteins. Each of the 'interesting' sentences, that has a potential gene or protein (as marked above), is further processed to identify words matching terms present in the interaction lexicon. If a match is found, the sentence is marked as a potential 'interaction' sentence. These are further disambiguated using the interactive tool, ATV, as described in Section 4.3.

Figure 4 shows an example query (search term MMP9) performed using BioFerret. Three search engines were selected for the query, namely Altavista, HotBot, and PubMed. Up to a maximum of 25 queries were fetched from each of the search engine. The retrieved results were merged to remove redundancies, and then ranked, clustered and classified. All documents classified as a publication are automatically processed further to identify potential entities and interactions of interest. In the current example, a set of user context files was used to describe the user's interest. These include interaction lexicons (for example, the 'BIND' lexicon consists of terms such as attach, assemble, bind, conjugate, createbond, etc.) and a list of nouns of interest (for example the 'SCI' context consists of genes, proteins, and disease specific terms related to spinal cord injury that the user is interested in). The user context files are used to determine relevancy of each of the search results and for identification of potential entities and interactions of interest. The search results are displayed in an HTML page in descending order of the computed relevancy. Results are also organized in terms of each user context and can be viewed by clicking on the relevant context at the top of the display. The user can select a specific article of interest, which is then displayed in detail in the right hand panel, as shown in Figure 4.

4.3 User-guided extraction

Automatic text analysis suffers from a number of limitations, as described in Section 2. Anaphora resolution, coordination resolution, and apposition resolution are still unsolved problems. Moreover, English language has inherent ambiguities. We therefore, rely on automated methods only to identify interesting and relevant information, and leave the final ambiguity resolutions to the user. We use user context, as described

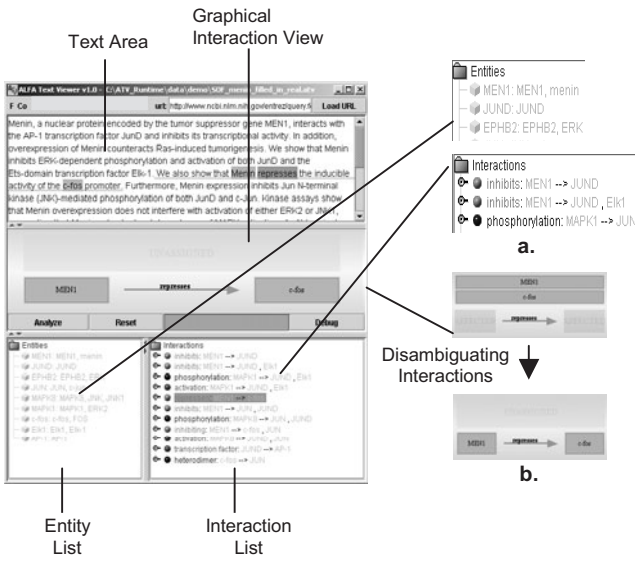

Fig. 5. A screenshot of ALFA Text Viewer (ATV); (a) an expanded view of the Entities and Interactions panel; (b) demonstrating the operation of disambiguating an interaction.

in Section 4.2, to identify potential gene and protein names and their interactions quickly from a large corpus of text. We then provide users with the ability to disambiguate the gene and protein symbols, resolve aliases, disambiguate the interactions, and encode the results as ALFA objects, via the ALFA Text Viewer (ATV).

Text documents from a corpus (search results) identified as relevant to the user based on relevance ranking in BioFerret are fed into ATV. ATV consists of a text window, a diagrammatic canvas, and two list-based editors. Figure 5 shows a screen shot of ATV. Arbitrary pieces of text or entire documents can be dragged or pasted into the text window. Potential entities (gene or protein symbols identified via BNS lookup or other nouns defined in user context files) and their potential interactions are identified using the methods described in Section 4.2 and displayed in the two list- based editors. Automatic linking is provided between the entities and interactions in list-based editors and their occurrence in text, via back-pointers to text. All occurrences of an entity are highlighted in the text, when that entity is selected in the list-based editor. Similarly, when an interaction is selected in its list-based editor, the sentence in the text and the potential entities involved in the interaction are highlighted in the text. Each interaction also highlights the potential entities involved and their potential roles. Initially, all potential entities in an interaction are assigned the 'unassigned' role. The user can drag entities into 'upstream', 'downstream', and 'mediator' roles to disambiguate the directionality of the interaction. The diagrammatic canvas displays a graphical representation of an interaction. While the user can disambiguate interactions by dragging and dropping entities

into specific roles in the list-based interaction editor, the user can also work in the equivalent representation in the diagrammatic canvas. All the windows are synchronized, such that any change in one window is synchronously displayed in the other windows. The tool further provides a simple user interface to add, delete, and modify entities and interactions interactively. Selected list of entities and interactions can be saved as ALFA objects.

4.3.1 Alias Management One of the major limitations to automatic identification of gene and protein names is the lack of a standard naming convention (Wong, 2001). It is not uncommon for a gene or protein to be known by a number of different aliases. ATV offers a simple alias management tool, which relies on BNS (Kincaid *et al*., 2002) and user context files. Symbols identified via BNS are automatically converted to the formal name as referred to in LocusLink. Moreover, all the aliases are automatically mapped to the same formal name. The user can interactively modify or update the name of an entity, if it does not occur in LocusLink or if multiple matches are found. Further, the user context files can also define entities and their aliases. In our alias resolution strategy, user context entries take precedence over the LocusLink entries.

4.3.2 Conversion to ALFA Object Model Entities and interactions disambiguated by the user are converted to ALFA objects by both of our text processing tools, BioFerret and ATV. In case of BioFerret, since only potential entities and interactions are identified (without performing any verification or disambiguation), the underlying ALFA relation objects are populated with 'unknown' roles and many of the property fields of the ALFA concept and relation objects are not set. However, if the user has disambiguated the entities and relations via ATV, then the respective ALFA relation objects contain defined roles, such as upstream, downstream, etc. In both the cases, the source link (text document and sentence) is also attached as a property of the ALFA concept and relation objects. If multiple documents refer to the same relation or concept, the source link property of the respective ALFA object is modified to contain the list of text sources. Thus, ALFA objects always maintain a pointer to the original source for later re-use, collaborative sharing, or verification. Multiple relations for the same concepts can be created in ALFA, thus allowing easy management of ambiguities and contradictions, which commonly exist in scientific literature.

5 APPLICATIONS

Our motivation for developing the ALFA architecture is based on collaborations with pharmaceutical and academic researches and identifying their needs in integrating multiple heterogeneous data sources for analyzing high-throughput experimental data, synthesizing knowledge from these data, and storing this knowledge for re-use later or for collaborative sharing. We next describe the potential of the ALFA architecture to aid biologists in their drug discovery research.

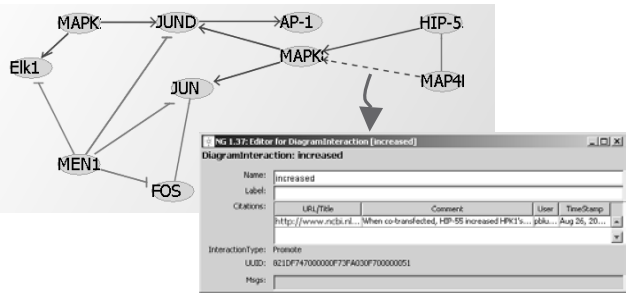

Fig. 6. A screenshot of the ALFA Network Viewer (ANV); the pop up demonstrates the link to the original document and sentence describing the selected relation.

5.1 Visualization

The ALFA object model easily lends itself to a visual graphical representation as shown in Figure 3. ALFA relation objects map to a hyper-graph, with ALFA role and node objects mapping to the edges and nodes in the hyper-graph. We are developing an ALFA Network Viewer (ANV) tool for interactive visualization of the ALFA objects. The tool is capable of displaying graphics consisting of static images (gif, jpeg, or bitmap format) or manually constructed network diagrams. Users can manually construct a network diagram either independently or by loading a pre-existing image and use it as a background template to sketch over. The tool further converts the constructed network diagrams to respective ALFA object representations. Figure 6 shows a screen shot of an ALFA network graphically displayed in the ANV.

5.2 Navigating text corpus

We are developing a novel scheme for displaying a visual summary of a corpus of text documents (say a set of search result documents) based on the identified ALFA concepts and relations. The corpus of text can be generated through a query using the meta-search engine, BioFerret, or processed through the ATV tool. The resultant extracted ALFA objects (which have links to the text source) are then displayed in the ANV as a visual summary of the text corpus. By selecting a concept or a relation, the user can easily navigate to the specific location in a textual document where the concept or the relation was found. Further, the ALFA objects can be used to filter the corpus. For example, the corpus can be filtered with respect to all relations containing a specific concept, say MMP9. The visual interface thus, serves as a navigation and filtering tool to sift through the text corpus. In other words, the network of relations serves as a visual Table of Contents for the text corpus.

5.3 Comparing disparate information

Due to its standardized representation of information, the ALFA architecture allows easy comparisons of information

extracted from heterogeneous sources of data. ALFA networks extracted out from scientific text can be computationally compared with networks extracted from known pathway databases (such as KEGG). The underlying ALFA models from the two sources can be computationally compared to visually highlight the similarities and differences between the two models via well-known visualization techniques such as overlays, coloring, or highlighting.

5.4 Extending network diagrams

BioFerret and ATV can also be used in conjunction with ANV to extend pre-existing biological models. The user loads a pre-existing ALFA network model and uses BioFerret to run a batch query with each of the concepts in the current model. The retrieved results are filtered to return only those relations that have at least one concept from the original pre-existing model. These relations are then integrated into the pre-existing network in a two-step process. First, each of the filtered relations is added to the pre-existing network. Note that at this stage, relations are not joined together by nodes. Next, all new relations are joined at the common concept to the pre-existing relations. In other words, the nodes for the concept are combined to form a single node.

5.5 Microarray data analysis

Computational methods for analyzing microarray data find statistical correlations, i.e., a set of up- or down-regulated genes for specific conditions, and not causal relationships, which biologists are most interested in. A recent study (Clare and King, 2002) discusses the problem that most genes correlated to disease mechanisms as identified by microarray studies do not necessarily cluster together in terms of their GO functional classifications. Therefore, biologists are interested in finding how these sets of up- and down-regulated genes are related to the disease mechanism. Further, they are interested in the pathways that these genes are involved in and whether these pathways are involved in the disease mechanism under study.

The ALFA architecture can extract information specific to a set of genes and disease mechanisms and relate these to the microarray experimental data. Given a set of interesting genes (say up- and down-regulated genes from a microarray experiment), BioFerret first automatically retrieves a large corpus of documents for all pairwise queries between the sets of genes (and their aliases) and also for queries between the genes in the set and concepts (user context) defining the disease mechanism of interest. For example, if genes MMP9 and MMP1 were up-regulated and the disease context is SCI or spinal cord injury, the following queries will be run on BioFerret: (i) (MMP9 OR GELB OR CLG4B) AND (MMP1 OR CLG OR CLGN); (ii) (MMP9 OR GELB OR CLG4B) AND (SCI OR other disease mechanism specific context terms); (iii) (MMP1 OR CLG OR CLGN) AND (SCI OR other disease mechanism specific context terms). The retrieved documents are then filtered based on the user context and BNS (Kincaid *et al*., 2002) to identify sentences that define interactions between these genes or between the genes and the disease mechanism. Given a set of 500 genes and a limit on fetching 10 documents per query, the entire automated process of query formulation, search, filtering, and identifying relevant interactions from PubMed using BioFerret takes approximately 30 minutes on a Pentium II processor with 800 MHz clock speed, 1GB RAM, and running Windows 2000 as the operating system. Most of the time (approximately 75%) is spent on retrieving documents (abstracts) from the PubMed database.

The list of potential interactions can be either visualized directly using ANV or first disambiguated using ATV and then visualized in ANV. A priori knowledge can be incorporated into ALFA via pre-existing pathway diagrams from say KEGG or manually crafted ALFA network diagrams. Textual information can be brought in to extend the existing knowledge base as described in Section 5.4. Moreover, the knowledge can be updated based on the inconsistencies between existing knowledge and newer information extracted from text. Such inconsistencies and discrepancies can be easily and automatically highlighted in the ANV using the ALFA architecture. Finally, experimental data can be overlaid from AEV (e.g., use of different colors for up- and down-regulated genes) on top of the constructed diagrams for visual analysis.

6 CONCLUSIONS AND FUTURE WORK

There is a critical need for heterogeneous data and information extraction, integration and management tools for knowledge discovery and synthesis in life science research. Scientific text is an important source of pre-existing biological knowledge from which biologists want to extract information relevant to their research. However, automatic extraction of information from scientific text suffers from a large number of drawbacks. The intended end users do not necessarily understand the computational processes involved in the automated extraction and they have a hard time trusting the output. It is hard to quantify the accuracy of automated methods in a general-purpose extraction task. Moreover, automated extraction does not take into account a user's context, since these processes are static and not user-trainable. For example, automated techniques yield the same result to different users seeking different bits of information from the same document corpus. Finally, extraction of information from text is not necessarily an end in itself. Biologists need to relate this information to other sources of information, such as experimental data or diagrammatic biological models.

We have therefore, proposed an architecture (ALFA) consisting of a suite of integrated software tools for interactive, user-guided extraction of information from heterogeneous data sources and representation of the extracted information in a structured format.

The representation of information in a structured manner aids in automatic linking, transformation, overlaying, and comparison of information extracted from heterogeneous data sources. Specifically, we describe a set of software tools for user-guided extraction of information from scientific text and linking of this information via the structured format to other sources of biological information (such as diagrammatic biological models and experimental data). We have developed a meta-search tool (BioFerret) for information retrieval from text, which uses 'user context' to filter, identify, and sift through the retrieved documents for relevant information. An interactive tool (ATV) further aids the user to guide the information extraction process, allowing the user to easily select interesting text, disambiguate the relationships described in the text, and capture it in the structured format. These tools can also be applied in the batch mode (over a corpus of text or over results combined from multiple searches) to speed the extraction process. While interactive information extraction is potentially slower than automatic processing, we feel that the advantages lie in the flexibility of our tools in applications to a diverse number of information extraction problems, the easy to use and understand nature of these tools, their ability to provide the user the control over the extraction process, and their ability to transform the information extracted from heterogeneous sources of data to a structured computational format. As part of our future work, we are developing the visualization tools for biological diagrams and experimental data (ANV and ATV) and integrating them into the ALFA architecture. We are also closely working with a set of research scientists in academia and pharmaceutical industry to apply these tools to aid in the analysis of their high throughput genomic and proteomic data.

ACKNOWLEDGEMENTS

The authors thank David Moh and Christian LeCocq for their inputs in the design of the ALFA Object Model, and Will Old and Christian LeCocq for valuable discussions in defining the base ontology.

REFERENCES

Clare,A. and King,R.D. (2002) How well do we understand the clusters found in microarray data? *InSilico Biol.*, **2**, 511–522.

Collier,N., Nobata,C. and Tsujii,J. (2000) Extracting the names of genes and gene products with a hidden Markov model. In *Proceedings of the 18th International Conference on Computational Linguistics, Universität des Saarlandes*, Saarbrücken, Germany, July 31–August 4. Association for Computational Linguistics, Morristown, NJ, pp. 201–207.

Friedman,C., Kra,P., Yu,H., Krauthammer,M. and Rzhetsky,A. (2001) GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, **1**, 1–9.

Fukuda,K., Tsunoda,T., Tamura,A. and Takagi,A. (1998) Toward information extraction: identifying protein names from biological papers. *Pac. Symp. Biocomput.*, **3**, 705–716.

Gene Ontology™ (2004) Gene Ontology Consortium.

Humphreys,K., Demetriou,G. and Gaizauskas,R. (2000) Two applications of information extraction to biological science journal articles: enzyme interactions and protein structures. *Pac. Symp. Biocomput.*, **5**, 502–513.

Iliopoulos,I., Enright,A.J. and Ouzounis,C.A. (2001) TEXTQUEST: document clustering of MEDLINE abstracts for concept discovery in molecular biology. *Pac. Symp. Biocomput.*, **6**, 384–395.

Jain,A.K. and Dubes,R.C. (1998) *Algorithms for Clustering Data*. Prentice Hall, Englewood, NJ.

Kincaid,R., Kleusing,D. and Vailaya,A. (2002) BNS: an LDAP-based biomolecule naming service. In *Proceedings of the Conference on Objects in Bio- & Chem-Informatics 2002 (OiBC-2002)*, Arlington, VA, November 18–19.

Krauthammer,M., Rzhetsky,A., Morozov,P. and Friedman,C. (2000) Using BLAST for identifying gene and protein names in journal articles. *Gene*, **259**, 245–252.

Ng,S.-K. and Wong,M. (1999) Toward routine automatic pathway discovery from on-line scientific text abstracts. *Genome Inform.*, **10**, 104–112.

O'Day,V.L., Adler,A., Kuchinsky,A. and Bouch,A. (2001) When worlds collide: molecular biology as interdisciplinary collaboration. In Prinz,W., Jarke,M., Rogers,Y., Schmidt,K. and Wulf,V. (eds), *Proceedings of the Seventh European Conference on Computer Supported Cooperative Work*, Bonn, Germany, September 16–20. Kluwer Academic Publishers, Dordrecht, pp. 399–418.

Palakal,M., Mukhopadhyay,S., Mostafa,J., Raje,R., N'Cho,M. and Mishra,S. (2002a) An intelligent biological information management system. *Bioinformatics*, **18**, 1283–1288.

Palakal,M., Stephens,M., Mukhpodhyay,S., Raje,R. and Rhodes,S. (2002b) A multi-level text mining method to extract biological relationships. In *Proceedings of the 1st IEEE Computer Society Bioinformatics Conference (CSB 2002)*, Stanford, CA, August 14–16. IEEE Computer Society.

Park,J.C., Kim,H.S. and Kim,J.J. (2001) Bidirectional incremental parsing for automatic pathway identification with combinatory categorical grammar. *Pac. Symp. Biocomput.*, **6**, 396–407.

Porter,M.F. (1980) An algorithm for suffix stripping. *Program*, **14**, 130–137.

Rindflesch,T.C., Tanabe,L., Weinstein,J.N. and Hunter,L. (2000) EDGAR: extraction of and drugs, genes, and relations from the biomedical literature. *Pac. Symp. Biocomput.*, **5**, 514–525.

Sekimizu,T., Park,H.S. and Tsujii,J. (1998) Identifying the interaction between genes and gene products based on frequently seen verbs in medline abstracts. *Genome Inform.*, **9**, 62–71.

Stephens,M., Palakal,M., Mukhopadhyay,S., Raje,R. and Mostafa,J. (2001) Detecting gene relations from MEDLINE abstracts. *Pac. Symp. Biocomput.*, **6**, 483–496.

Wahl,M., Howes,T. and Kille,S. (1997) Lightweight Directory Access Protocol (v3). IETF RFC2551.

Wong,L. (2001) PIES, a Protein Interaction Extraction System. *Pac. Symp. Biocomput.*, **6**, 520–531.

Yakushiji,A., Tateisi,Y., Miyao,Y. and Tsujii,J. (2001) Event extraction from biomedical papers using a full parser. *Pac. Symp. Biocomput.*, **6**, 408–419.