Data Mining and Predictive Modeling of Biomolecular Network from Biomedical Literature Databases

Xiaohua Hu and Daniel D. Wu

Abstract—In this paper, we present a novel approach Bio-IEDM (Biomedical Information Extraction and Data Mining) to integrate text mining and predictive modeling to analyze biomolecular network from biomedical literature databases. Our method consists of two phases. In phase 1, we discuss a semisupervised efficient learning approach to automatically extract biological relationships such as protein-protein interaction, protein-gene interaction from the biomedical literature databases to construct the biomolecular network. Our method automatically learns the patterns based on a few user seed tuples and then extracts new tuples from the biomedical literature based on the discovered patterns. The derived biomolecular network forms a large scale-free network graph. In phase 2, we present a novel clustering algorithm to analyze the biomolecular network graph to identify biologically meaningful subnetworks (communities). The clustering algorithm considers the characteristics of the scale-free network graphs and is based on the local density of the vertex and its neighborhood functions that can be used to find more meaningful clusters with different density level. The experimental results indicate our approach is very effective in extracting biological knowledge from a huge collection of biomedical literature. The integration of data mining and information extraction provides a promising direction for analyzing the biomolecular network.

Index Terms—Biomolecular network, semisupervised learning, scale-free network, information extraction, biological complexes (communities).

1 Introduction

Despite an influx of molecular data in the form of sequences, structure, transcription profiles, etc., most of the protein interaction information relevant to cell biology research still exists strictly in the scientific literature, which is written in a natural language that computers cannot easily manipulate. A huge portion of the scientific literature (abstracts and/or articles) is collected in large online digital libraries such as PubMed, which is now estimated to contain more than 15 million abstracts. However, retrieving and processing this information is very difficult due to the lack of formal structure in the natural-language narrative in those documents and the huge size of the documents collected in the biomedical literature databases. Automatically mining and extracting information from biomedical text holds the promise of easily consolidating large amounts of biological knowledge in computer-accessible form. The development of reliable literature data mining technologies to maximally exploit data and information from this ever-expanding collection of scientific literature so that domain experts can analyze this information to form new hypotheses, conduct new experiments, and enable new discovery is essential in cell biology research.

A promising approach for making vast information manageable and easily accessible is to develop an information extraction (IE) system that automatically processes these documents, extracts important biological knowledge such as protein-protein interactions, functionality of the genes, subcellular location of the protein, etc., and consolidates them into databases. This serves several purposes: 1) It consolidates data about a single organism or a single class of entity (e.g., proteins, genes, etc.) in one place, making them very helpful for bioinformatics research at genomic scale in order to get a global view of that organism. 2) This process makes the information searchable and manageable since these results are extracted in a structured format. 3) The extracted knowledge can help researchers generate plausible hypotheses or at least clarify and classify biological knowledge so as to assist the user in generating hypotheses. It can also alter the user's perception of the biological relationships in such a way as to stimulate new experiments and methods. Some databases that accumulate these biological relationships are DIP for protein-protein interactions [47], KEGG for biological pathways [30], and BIND for molecular interactions [2]. The biological knowledge stored in these databases is almost entirely manually assembled. However, it is becoming more and more difficult for curators to keep up with the increasing volume of literature. Thus, automatic methods are needed to speed up this step of database construction. Integration of Web mining, text mining, and information extraction provides a promising direction to assist in the curation process to construct such databases.

• *The authors are with the College of Information Science and Technology, Drexel University, Philadelphia, PA 19104.*
E-mail: thu@cis.drexel.edu, daniel.wu@drexel.edu.

Manuscript received 17 Feb. 2006; revised 9 June 2006; accepted 15 Aug. 2006; published online 12 Jan. 2007.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBBSI-0021-0206.
Digital Object Identifier no. 10.1109/TCBB.2007.070211.

On the other hand, mining the biomolecular network to identify a community (subnetwork) has become a very hot topic because communities are believed to play a central role in the functional properties of complex networks [34]. Studying the community structure of biological networks is of particular interest and challenging, given the enormous number of genes and proteins and the complex nature of the interactions among them. In the context of biological networks, communities might represent structural or functional groupings and can be synonymous with molecular modules, biochemical pathways, gene clusters, or protein complex. Being able to identify the community structure in a biological network hence could help us to better understand the structure and dynamics of biological systems. Analysis of a number of metabolic networks from different organisms has revealed communities that relate to functional units in the networks [24]. Communities of related genes have been reported in a network of gene relationships as established by cooccurrence of gene names in the published literature [45]. In our previous work, we developed a spectral-based clustering method using local density and vertex neighborhood to analyze the chromatin network [28], [27], [46]. Two recent works along this line of research are based on the concept of network modularity introduced by Hartwell et al. [20]. The works by [37], [40] both used computational analyses to cluster the yeast protein-protein interaction network and discovered that molecular modules are densely connected with each other but sparsely connected with the rest of the network.

In this paper, we present a novel approach, Bio-IEDM, dealing with these two important issues in a unified method simultaneously. Bio-IEDM consists of two phases: 1) construction of the biomolecular network through information extraction from the biomedical literature database and 2) mining the biomolecular network to identify biologically meaningful subnetworks (communities). Bio-IEDM integrates information extraction, text mining, and predictive modeling to analyze a biomolecular network from biomedical literature databases.

The rest of the paper is organized as follows: In Section 2, we review the related work in biomedical information extraction and mining, biomolecular network analysis. In Section 3, we first present the architecture of BIO-IDEM, then discuss the construction of the biomolecular network through biomedical literature mining. We focus on the automatic query learning method for selecting promising text files from the text databases for extraction and the mutual reinforcement approach for pattern extraction and tuple extraction. Next, we present our novel algorithm, *CommBuilder*, to mine the entire biomolecular network to identify the biological relevant subnetwork (community) and the experimental results on two large biomolecular networks: the yeast interaction network and the chromatin network. We summarize our major contributions in Section 4.

2 Related Work

Biomedical literature mining from a biomedical database (mainly PubMed) has attracted a lot of attention recently from the information extraction, data mining, natural language understanding (NLP), and bioinformatics community [22], [19]. A lot of methods have been proposed and various systems have been developed for extracting biological knowledge from the biomedical literature, such as finding protein or gene names [15], [41], protein-protein interactions [5], [33], [12], [36], protein-gene interactions [9], subcellular location of protein, functionality of gene, protein synonyms [11], etc. For example, in their pioneering work in biomedical literature mining, Fukuda et al. [15] rely on special characteristics such as the occurrence of uppercase letters, numerals, and special endings to pinpoint protein names. Stapley and Benoit [41] extracted co-occurrences of gene names from MEDLINE documents and used them to predict their connections based on their joint and individual occurrence statistics. Blaschke et al. [5] propose an NLP-based approach to parse sentences in abstracts into grammatical units and then analyze sentences discussing interactions based on the frequency of individual words. Because of the complexity and variety of the English language, such an approach is inherently difficult. Ono et al. [36] manually defined some regular expression patterns to identify the protein-protein interactions. The problem is that regular expression searches for abstracts containing relevant words, such as "interact," "bind," etc., poorly discriminates true hits from abstracts using the words in alternative senses and misses abstracts using different language to describe the interactions. Their method relies on a manually created "pattern" to the biological relationship. This approach may introduce a lot of "false positives" or "false negatives" and it is unable to capture the new biological relationships not in those "manual" patterns. Marcott et al. [33] proposed a Bayesian approach based on the frequencies of discriminating words found in the abstracts. They score Medline abstracts for probability of discussing the topic of interest according to the frequencies of discriminating words found in the abstract. The highly likely abstracts are the sources for the curators for further examination for entry into the databases. Hahn et al. [18] developed the MEDSYNDIKATE based on NLP techniques to extract knowledge from medical reports. Although the approaches differ, they can all be seen as examples of this process: First, select what will be read, then identify important entities and relations between those entities, and, finally, combine this new information with other documents and other knowledge. These systems, however, suffer from various weaknesses. First, the templates these systems are supplied with allow only factual information about particular, a prior chosen entities (cell type, virus type, protein group, etc.) to be assembled from the analyzed documents. Also, these knowledge sources are considered to be entirely static. Accordingly, when the focus of interest of a user shifts to a topic not considered so far, new templates must be supplied or existing ones must be updated manually.

Networks have been used to model many real-world phenomena in bioinformatics to better understand the phenomena and to guide experiments in order to predict their biological behavior. It is very important to have a model in order to provide good guidance for the experiments. As a result, new techniques and models for analyzing and modeling real-world networks have recently

been introduced. Recently, empirical studies report that the protein-protein interaction network [20], [34], like many other network graphs generated either from the real world or the manmade world such as the Internet, the WWW, have scale-free properties [3], [34]. Scale-free networks have been used to explain behaviors as diverse as those of power grids, the stock market, and cancerous cells, as well as the biomolecular network. Put simply, the nodes of a scale-free network aren't randomly or evenly connected. Scale-free networks include many "very connected" nodes, hubs of connectivity that shape the way the network operates. The ratio of very connected nodes to the number of nodes in the rest of the network remains constant as the network changes in size. In a scale-free network, the nodes with the largest numbers of links play an important role on the dynamics of the system. The scale-free property reveals that the number of incoming links and the outgoing links at a given vertex have distributions that decay with the power law tails [34]. It is essential to mine the network graph to help understand the domain and the topology of the network structure. For example, a local cluster in a biological interaction network for proteins may represent a biological complex [16], which is very important to help understand the protein functionality.

Many graph-based clustering algorithms have been developed to analyze the network graphs [37] to identify communities or subnetworks (biological complexes). Although there is no formal definition for the community structure in a network, it often loosely refers to the gathering of vertices into groups such that the connections within groups are denser than between groups [17]. The study of community structure in a network is not new. It is closely related to the graph partitioning in graph theory and computer science and the hierarchical clustering in sociology [34]. However, recent years have witnessed intensive activity in this field, partly due to the dramatic increase in the scale of networks being studied. Many algorithms for finding communities in networks have been proposed. They can be roughly classified into two categories, divisive and agglomerative. The divisive approach takes the route of recursive removal of vertices (or edges) until the network is separated into its components or communities, whereas the agglomerative approach starts with isolated individual vertices and joins together small communities. One important algorithm is proposed by Newman (the GN algorithm) [34]. The GN algorithm is based on the concept of betweenness, a quantitative measure of the number of shortest paths passing through a given vertex (or edge). The vertices (or edges) with the highest betweenness are believed to play the most prominent role in connecting different parts of a network. The GN algorithm detects communities in a network by recursively removing these high betweenness vertices (or edges). It has produced good results and is well adopted by different authors in studies of various networks [34], but has a major disadvantage, which is its computational cost. For sparse networks with n vertices, the GN algorithm is of $O(n^3)$ time. Various alternative algorithms have been proposed [35], [13], [44], attempting to improve either the quality of the community structure or the computational efficiency of finding communities.

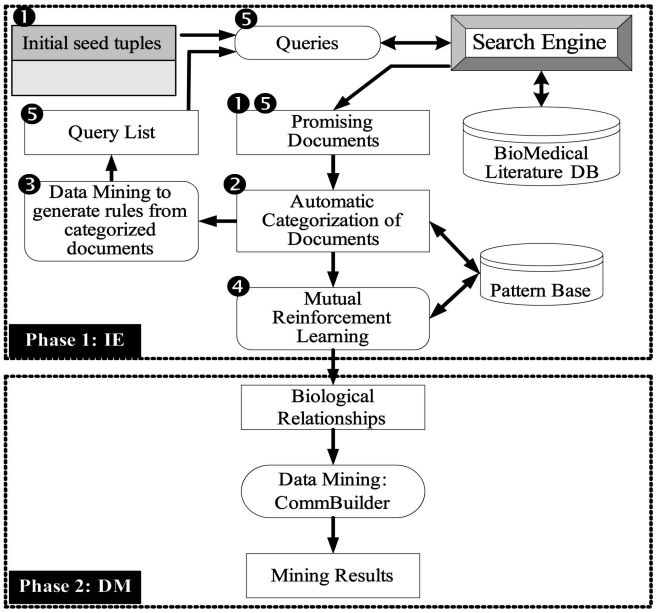


Fig. 1. Bio-IEDM.

3 The Architecture and Principle of BIO-IEDM

In this paper, we present a novel scalable, portable, and robust extraction and mining system from biomedical literature, Bio-IEDM (Biomedical Information Extraction and Data Mining), as shown in Fig. 1. It integrates information extraction and robust data mining to automatically extract and mine biological relationships from a huge collection of biomedical literature to help biologists in functional bioinformatics research.

Bio-IEDM consists of two phases: **Phase 1** (**IE**: Information Extraction): Bio-IEDM extracts the protein-protein interaction from the biomedical literature. These extracted protein-protein interactions form a scale-free network graph that has many distinct properties, such as the in-degrees and out-degrees of the vertices following power laws. In **Phase 2** (**DM**: Data Mining), we apply a clustering method, *CommBuilder*, to mine the protein-protein interaction network. The clusters in the network graph represent some potential protein complexes, which are instrumental to biologists in the study of the protein functionality. The details of Phases 1 and 2 are discussed in the sections below.

3.1 Construction of Biomolecular Network through Biomedical Literature Data Mining

We develop a semisupervised efficient learning approach to automatically extract biological knowledge from the biomedical literature databases to construct the biomolecular network. "Semisupervised learning" refers to the use of both labeled and unlabeled data for training. It contrasts supervised learning (data all labeled) or unsupervised learning (data all unlabeled). Standard classifier training uses **only** labeled data (feature/label pairs), not unlabeled data. However, labeled data is often hard to get because they need experienced human annotators, while unlabeled data may be relatively easy to collect. The goal of semisupervised learning

TABLE 1
Initial Training Seed Tuples

Protein 1	Protein 2	Interaction
HP1	histoneH3	Yes
HP1	HDAC4	Yes
KAP1	SETDB1	Yes
AuroraB	INCENP	Yes

is to train better classifiers from both labeled and unlabeled data [6].

Our method extracts biological knowledge from biomedical libraries and requires only a handful of training examples from users. These examples are used as seed tuples to generate extraction patterns that in turn result in new tuples being extracted from the biomedical literature database. It consists of the following steps (the number below corresponds to the number in Fig. 1):

1. Starting with a set of user-provided seed tuples (the seed tuples can be quite small), our system retrieves a sample of documents from the biomedical digital library. At the initial stage of the overall document retrieval process, we have no information about the documents that might be useful for the goal of extraction. The only information we require about the target relation is a set of user-provided seed tuples, including the specification of the relation attributes to be used for document retrieval. We construct some simple queries by using the attribute values of the initial seed tuples to extract document samples of a predefined size using the search engine.
2. The tuple set induces a binary partition (a split) on the documents: those that contain tuples or those that do not contain any tuple from the relation. The documents are thus labeled automatically as either positive or negative examples, respectively. The positive examples represent the documents that contain at least one tuple. The negative examples represent documents that contain no tuples.
3. We next apply data mining algorithms to derive queries targeted to match—and retrieve—additional documents similar to the positive examples.
4. Generate extraction patterns and extract new tuples based on pattern matching.
5. Query the biomedical digital library using the learned queries from Step 3 to retrieve a set of promising documents form the databases. Then, we go to Step 2. The whole procedure repeats until no new tuples can be added in the relation or we reach the preset limit of a maximal number of text files to process.

The details of the key steps are discussed in the subsequent sections

3.1.1 Learning Queries to Retrieve Potential Promising Biomedical Documents

Previous approaches for addressing the high computational cost of information extraction resorted to document filtering to select the document that deserved further processing by the information extraction system. This filtering technique still requires scanning the complete database to consider every document. Alternative approaches use keywords or phrases as filter (which could be converted to queries) that were manually crafted and tuned by the information extraction system developers. In the biomedical and bioinformatics domain, there exist research topics that cannot be uniquely characterized by a set of key words because relevant keywords are 1) also heavily used in other contexts and 2) often omitted in relevant documents because the context is clear to the target audience. Information retrieval interfaces such as entrez/PubMed produce either low precision or low recall in this case.

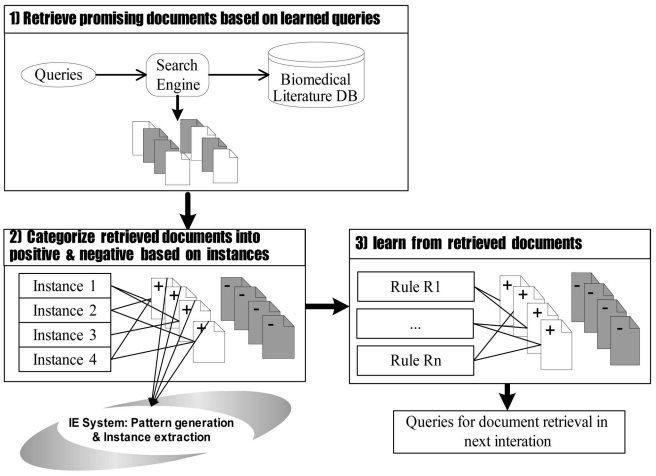


Fig. 2. The process of learning queries from retrieved documents.

To yield a high recall at a reasonable precision, the results of a broad information retrieval search have to be filtered to remove irrelevant documents. We use automated text categorization for this purpose. In the initial round, we select a prespecified number of documents based on the seed examples. For example, if our system is used for extracting protein-protein interactions, the seed examples are a set of protein name pairs, as shown in Table 1. So, we can first select all of those documents in PubMed which contain all of those protein names in the seed examples. If a document does contain the seed examples in a single sentence, we label it as a positive example; otherwise, it is negative. These labeled documents are used in the later stage by a data mining algorithm to learn the characteristics of the documents. The learned rules are converted to a query list in order to retrieve potentially promising documents for IE in the next iteration. Starting from the second round, we use the query list derived from the learned rules to select potential interesting documents and rely on all the available tuples for document classification. The process is illustrated in Fig. 2.

Our approach automatically discovers the characteristics of documents that are useful for extraction of a target relation, starting with only a handful of user-provided examples of tuples of the relation to extract. Using these tuples as seeds, our system retrieves a sample of documents from the database. By running the information extraction system over the documents, we identify which documents are useful for the extraction task at hand. Then, we apply data mining techniques to learn queries that will tend to

match additional useful documents. Given a set of useful and useless documents as the training set, our goal now is to generate queries that would retrieve many documents that the IE system will find useful and few that IE will not be able to use. The process consists of two stages: 1) convert the positive and negative examples into an appropriate representation for training and 2) run the data mining algorithms on the training examples to generate a set of rules and then convert the rules into an ordered list of queries expected to retrieve new useful documents. In our current implementation, we integrate three algorithms: Ripple [10], CBA [32], and our own maximally generalized decision rules DB-Deci [25]. We rank all of the rules based on the Laplace measures and the top 10 percent of the rules is converted into a query list [8]. Using Laplace measure, many rules covering a few examples are eliminated as the significance test believes their apparent high accuracy is likely to be simply due to chance.

For example if a rule set is

```
Positive IF WORDS have protein AND binding,
Positive IF WORDS have cell and function,
```

then it can convert the rule set to a query list

```
Query 1: protein AND binding
Query 2: cell AND function
```

Unlike most other IR system which use a single term selected with a statistically-based term weighting, we use a data mining algorithm to extract rules from the documents and then use the terms from the rules as the basic unit for our query term.

3.2 Mutual Reinforcement Principle for Pattern Generations and Tuple Extraction

A crucial step in the extraction process is the generation of new patterns, which is accomplished by grouping the occurrences of known patterns in documents that occur in similar contexts. A good pattern should be selective but have high coverage so that they do not generate many false positives and can identify many new tuples. Most machine learning methods and algorithms that have been developed to automatically generate extraction patterns use special training resources, such as texts annotated with domain-specific tags (e.g., AutoSlog [38] and WHISK [39]). A key limitation of using machine learning methods to induce IE methods is the requirement of having high-quality preclassified corpora in information extraction from a text database. Creating a preclassified corpus entails a high workload for domain experts and a corpus for a specific domain cannot usually be directly transferred to other domains, thus making portability a very challenging issue. Another bottleneck of machine learning approaches to learn patterns is that most learning algorithms rely on feature-based representation of objects. That is, an object is transformed into a collection of position-independent features $f_1, f_2, \ldots, f_n$, (f_i can be an n-gram word in the document), thereby producing an N-dimensional vector (also known as bag-of-word representation). The limitation of this representation is that, in many cases, data cannot be easily expressed via features. For example, in most NLP problems, feature-based representations produce inherently local representations of objects as it is computationally infeasible to generate features involving long-range dependencies. Kernel methods [42], [48] and relational learning are an attractive alternative to feature-based representations. One practical problem in applying kernel methods or relational learning to IE in large text collection is their speed. The two approaches are relatively slow compared to feature classifiers, whose computation complexity may be too high for practical purposes. The heart of our approach is a mutual reinforcement technique that learns extraction patterns from the tuples and then exploits the learned extraction patterns to identify more tuples that belong to the relation.

Our pattern representation uses Eliza-like patterns [43] that can make use of limited syntactic and semantic information. BIO-IEDM represents the context around the related entities in the patterns in a flexible way that produces patterns that are selective, yet have high coverage.

Definition 1. *A pattern is a 5-tuple*

$$\langle prefix, entity\_tag1, infix, entity\_tag2, suffix\rangle,$$

where $prefix$, $infix$, and $suffix$ are vectors associated weights with the terms. $Prefix$ is the part of sentence before $entity1$, $infix$ is the part of sentence between $entity1$ and $entity2$, and $suffix$ is the part of sentence after $entity2$.

For example, a protein-protein interaction pattern in our approach is a tuple (or expression) consisting of two protein names that correspond to some conventional way of describing interaction. We can use these patterns to characterize those sentences that capture this knowledge. For every such protein pair tuple $\langle p1, p2\rangle$, it finds segments of text in the sentences where $p1$ and $p2$ occur close to each other and analyzes the text that "connects" $p1$ and $p2$ to generate patterns. For example, our approach inspects the context surrounding chromatin protein *HP1* and *HDAC4* in *"HP1 interacts with HDAC4 in the two-hybrid system"* to construct a pattern $\{" ", \langle Protein\rangle, "interacts\ with, "\langle Protein\rangle, " "\}$. After generating a number of patterns from the initial seed examples, our system scans the available sentences in search of a segment of text that matches the patterns. As a result of this process, it generates new tuples and uses them as the new "seed" and starts the process all over again by searching for these new tuples in the documents to identify new promising patterns.

In order to learn these patterns from these sentences, we use a sentence alignment method to group similar patterns together and then learn each group separately for the generalized patterns.

Definition 2. *The $Match(T_i, T_j)$ between two 5-tuples*

$$T_i = \langle prefix_i, tag_{i1}, infix_i, tag_{i2}, suffix_i\rangle$$

and

$$T_j = \langle prefix_j, tag_{j1}, infix_j, tag_{j2}, suffix_j\rangle$$

is defined as

$$Match(T_i, T_j) = W_{prefix}{}^{*}Sim(prefix_i, prefix_j) + W_{infix}{}^{*}Sim(infix_i, infix_j) + W_{suffix}{}^{*}Sim(suffix_i, suffix_j).$$

There are many methods or formulas available to evaluate the similarity of two sentence segments such as $perfix_i$ and $prefix_j$, which are ordered lists of words, numbers, and punctuation marks, etc. In our system, we use a sentence alignment function similar to the sequence alignment in bioinformatics as shown in (3.1). The advantage of using sentence alignment for similarity measurement is that it is flexible and can be implemented efficiently based on dynamic programming. The same idea is also used in comparing the similarity between protein or DNA sequences. Given two sentence segments $X = (x_1, x_2, \ldots, x_m)$ and $Y = (y_1, y_2 \ldots, y_n)$, the similarity score $Sim(i, j)$ is defined as the score of the optimal alignment between the initial segment from x_1 to x_i of X and the initial segment from y_1 to y_j of Y. ($''-''$**denotes a white space**, $\mathbf{Sim(i, 0) = 0, Sim(0, j) = 0}$),

$$Sim(i,j) = \max\left\{\begin{array}{l} 0, \\ Sim(i-1, j-1) + f(x_i, y_j) \\ Sim(i-1, j) + f(x_i, ''-'') \\ Sim(i, j-1) + f(''-'', y_j) \end{array}\right\}, \quad (3.1)$$

$$f(x_i, y_j) = \log\frac{p(x_i, y_j)}{p(x_i) * p(y_j)}, \quad (3.2)$$

where $p(x_i)$ denotes the appearance probability of word x_i and $p(x_i, y_j)$ denotes the probability that x_i and y_j appear at the same position in two text segments. For sentence segment X with a length of m and Y with a length of n, in total $(m+1) * (n+1)$ scores will be calculated by applying (3.1) recursively. Store the scores in a matrix as $S = Sim(x_i, y_i)$. Through back-tracing in S, the optimal local alignment can be searched.

After generating patterns, Bio-IEDM scans the text collection to discover new tuples. Bio-IEDM first identifies sentences that include a pair of entities. For a given text segment, with an associated pair of entities E_1 and E_2, it generates the 5-tuples

$$\mathrm{T} = \langle perfix, E_1\_tag1, infix, E_2\_tag2, suffix\rangle.$$

A candidate tuple $\langle E_1, E_2\rangle$ is generated if there is a pattern T_p such that $Match(T, T_p)$ is greater than the prespecified threshold. Each candidate tuple will then have a number of patterns that helped generate it, each with an associated degree of match. Our approach relies on this information, together with score of the patterns (the score reflects the selectivity of the patterns), to decide what candidate tuples to actually add to the biological relationship table that is being constructed. Below are some sample extraction patterns generated from PubMed for protein-protein interaction.

$$\{'''', \langle Protein\rangle, ''interacts\ with, ''\langle Protein\rangle, ''''\}$$
$$\{'''', \langle Protein\rangle, ''binds\ to, ''\langle Protein\rangle, ''''\}$$
$$\{''Bind\ of, ''\langle Protein\rangle, ''to, ''\langle Protein\rangle, ''''\}$$
$$\{''Complex\ of''\langle Protein\rangle, ''and, ''\langle Protein\rangle, ''''\}.$$

Our method represents the context around the proteins in the patterns in a flexible way that produces patterns that are selective, flexible, and have high coverage. As a result, BIO-IEDM will ignore those minor grammar variations in the sentences and focus on the important key phases in the sentences.

Evaluation of Patterns and Tuples. Since there is no human feedback about the extracted tuples and patterns in this procedure, it is very important that the patterns and tuples generated during the extraction process be evaluated, bogus patterns be removed, and only highly selective and confident tuples be used as seed examples in the next iteration to ensure the high quality of patterns and tuples generated in each step. This way, our system will be able to eliminate unreliable tuples and patterns from further consideration.

Generating good patterns is challenging. For example, we may generate a pattern

$$\{'', ''\langle Protein\rangle, \langle''-''\rangle, \langle Protein\rangle\langle Interaction\rangle\}$$

from sentence "*these data suggest that the histoneH3-histoneH2b interaction is* ...." This pattern will be matched by any string that includes a protein followed by a hyphen, followed by another protein followed by the word "interaction." Estimating the confidence of the patterns so that we don't trust patterns that tend to generate wrong tuples is one of the problems that we have to consider. The confidence of the tuple is defined based on the selectivity and the number of the patterns that generate it. Intuitively, the confidence of a tuple will be high if it is generated by many highly selective patterns and a highly selective pattern tends to generate high confidence tuples This philosophy is similar the extraction of patterns and relations from the Web [7]. This idea is also similar to the concepts of hub and authoritative pages in Web searching [31].

We use a metric originally proposed by Riloff to evaluate extraction pattern $\mathrm{P_i}$ generated by the Autoslog-TS [38] in the information extraction system and define score $(\mathrm{P_i})$ as

$$Score(P_i) = F_i/N_i\ ^{*}\log(F_i), \quad (3.3)$$

where F_i is the number of unique tuples among the extractions produced by P_i and N_i is the total number of unique tuples that P_i extracted. This metric can identify not only the most reliable extraction patterns but also patterns that will frequently extract relevant information (even if irrelevant information will also be extracted).

For each tuple T_j, we store the set of patterns that produce it, together with the measure of similarity between the context in which the tuple occurred and the matching pattern. Consider a candidate tuple T_j and the set of patterns $P = \{P_i\}$ that were used to generate T_j. The confidence of an extracted tuple T_j is evaluated as

TABLE 2
Actual Patterns Discovered by Bio-IEDM

Confidence	Left	Middle	Right
0.82	""	Associate with	""
0.79	Bind of	to	""
0.75	""	-	complex
0.74	Interaction of	With	""

$$Conf(T_j) = 1 - \prod_{k=1}^{m}(1 - score(P_i) * Match(T_j)), \qquad (3.4)$$

where m is the number of patterns to generate T_j.

Thus, in the formulas above, $Conf(Pi)$ is not simply the count of the relevant tuples, but is rather their *cumulative relevance*. The two formulas, (3.3) and (3.4), capture the mutual dependency of patterns and tuples. This recomputation and growth of precision and relevance scores is at the heart of the procedure.

After determining the confidence of the candidate tuples using the definition above, our method discards all tuples with low confidence because these low quality tuples could add noise into the pattern generation process, which would in turn introduce more invalid tuples, degrading the performance of the system (in our experiment, the confidence threshold value is set to 0.7). For illustration purposes, Table 2 lists four representative patterns that our system extracted from the document collection.

3.2.1 Experiments

The goal of our system is to extract as much valid biological knowledge as possible from the huge collection of biomedical literature and to combine them into a database. We realize that a biological relationship may appear in multiple times in various documents, but we do not need to capture every instance of such relationships. Instead, as long as we capture one tuple of such a relationship, we will consider our system to be successful for that relationship. Evaluating the precision and recall of our BIO-IEDM system is very difficult because of the large collection of the documents involved. It is possible to manually inspect them and calculate the precision and recall for small biomedical documents sets. Unfortunately, this evaluation approach does not scale and becomes infeasible for a large collection of literature such as PubMed. Developing accurate evaluation metrics for this task is one of our future research plans. In this study, we conducted two experiments. One is to simulate the biologist manually creating a set of key word filters to select the documents which are relevant to protein interaction and then run the information extraction procedure on these documents to extract the protein-protein interaction pair (PPI). Nowadays, this is the approach used by the users of Medline. However, information retrieval in such databases can become very time-consuming because searchers that are likely to identify much relevant information also find many irrelevant documents. For example, a text query for "protein interaction" of the Medline database retrieves 196,960 documents (in January 2006). In this study, we use 1,600 human chromatin protein names provided by domain expert Professor Lechner from the Biological Science and Technology Department at Drexel University. Synonyms are derived from LocusLink and nucleotide databases maintained by NCBI. The total number of protein names is around 7,000. The key word list is manually constructed with the help of Prof. Lechner. The result is shown in Table 3. In our second experiment, we start with 10 pairs of protein-protein interaction pairs and use

TABLE 3
Number of PubMed Abstract Used in Our Test (Key Word-Based)

Keywords	# of abstracts	# of distinct PPI
Protein Associate	8025	760
Protein Interact	33835	2158
Protein Bind	69981	2664
Protein Association	82767	2093
Protein Binding	83397	3184
Protein interaction	145857	3795
Protein complex	185157	4300
Protein acetylate	172	116
Protein acetylation	5027	827
Protein conjugate	18770	92
Protein destabilize	879	31
Protein destabilization	2233	62
Protein inhibit	124178	1602
Protein modulate	41727	945
Protein modulation	71159	913
Protein phosphorylate	3991	315
Protein phosphorylation	90475	2249
Protein regulate	58586	2121
Protein regulation	289940	5915
Protein stabilization	27349	340
Protein stabilize	5714	221
Protein suppress	20069	633
Protein target	74714	2433
Total	1,444,002	37,769
Total (elimination of redundant ones)	1,006,699	9980

TABLE 4
Experimental Results (BIO-IEDM)

# of abstracts	# of PPI	# of distinct PPI
50k	2224	1749
100k	4412	3100
150k	8348	4400
200k	10527	5300
250k	12461	6040
300k	15152	6500
350k	16612	7200
400k	18202	8420
450k	19070	8900
all	19461	9183

Bio-IEDM to automatically construct queries and use the learned queries to retrieve documents from PubMed. In each iteration, we set the maximum document size to 10k for each iteration, starting with 50,000 documents and stopping at 500,000 documents when the number of new tuples added is very small (10 in our experiment). We repeat the experiments five times with different seed-pairs and take the average number of documents as the results, as shown in Table 4. We repeat the same procedure for the yeas gene interaction (because of space limitations, the search result of the yeast interaction network is omitted). The two biological relationship networks, chromatin network and yeast interaction network, extracted by our system Bio-IEDM are furthered analyzed using data mining algorithm *CommBuilder* to find potential biological complexes, as described in Section 3.2.

It is obvious that Bio-IEDM has a significant performance advantage over the key word-based approach. IBIO-IEDM only examined 500 K abstracts from PubMed to extract 9,183 distinct chromatin protein-protein interaction pairs, while the key word-based approach examined 1.4 million abstracts from PubMed to extract 9,980 distinct chromatin protein-protein interaction pairs. The result from Bio-IEDM has an overlap of 92 percent with the total protein-protein interaction pairs from the key word-based approach, while Bio-IEDM only searched around 1/3 of the total abstracts used in the key word-based approach.

3.3 Mining the Scale-Free Protein-Protein Interaction Network

In this paper, we address a basic question about the community structure in protein-protein interaction network, i.e., what community does a given protein (or proteins) belong to. Due to the complexity and modularity of biological networks, it is more feasible computationally to study a community containing a few dozen proteins of interest. Hashimoto et al. [21] have used a similar approach to growing genetic regulatory networks from seed genes. Their work is based on probabilistic Boolean networks and subnetworks are constructed in the context of a directed graph using both the coefficient of determination and the Boolean function influence among genes. A similar approach is also taken by Flake et al. [14] to find highly topically related communities on the Web based on the self-organization of the network structure and on a maximum flow method. Our approach, however, takes full advantage of the underlying topological properties of the networks.

3.3.1 The Algorithm CommBuilder

We intuitively model the protein-protein interaction network as an undirected graph, $G = (V, E)$, where vertices V represent proteins and edges E represent interactions between pairs of proteins. The graphs we use in this paper are unweighted and simple—meaning no self-loops or parallel edges.

For a subgraph $G' \subset G$ and a vertex i belonging to G', we define the in-community degree for vertex i, $k_i^{in}(G')$, to be the number of edges connecting vertex i to other vertices belonging to G' and the out-community degree, $k_i^{out}(G')$, to be the number of edges connecting vertex i to other vertices that are in G but not in G'. We adopt the quantitative definitions of community defined by [37], i.e., the subgraph G' is a community in a strong sense if $k_i^{in}(G') > k_i^{out}(G')$ for each vertex i in G' and in a weak sense if the sum of all degrees within G' is greater than the sum of all degrees from G' to the rest of the graph.

The algorithm, called *CommBuilder*, accepts the seed protein s, gets the neighbors of s, finds the core of the community to build, and expands the core to find the eventual community. The two major components of *CommBuilder* are *FindCore* and *ExpandCore*. Basically, *Find Core* performs a naive search for maximum clique from the neighborhood of the seed protein by recursively removing vertices with the lowest in-community degree until all vertices in the core set have the same in-community degree.

The algorithm performs a breadth first expansion in the core expanding step. It first builds a candidate set containing the core and all vertices adjacent to each vertex in the core (Step 16). It then adds to the core a vertex that either meets the quantitative definition of community in a strong sense or the fraction of in-community degree over a relaxed affinity threshold f of the size of the core (Step 21). The affinity threshold is 1 when the candidate vertex connects to each of vertices in the core set. This threshold provides flexibility when expanding the core because it is too strict, requiring every expanding vertex to be a strong sense community member.

The *FindCore* is a heuristic search for a maximum complete subgraph in the neighborhood N of seed s. Let K be the size of N, then the worst-case running time of *FindCore* is $O(K^2)$. The *ExpandCore* part costs, in the worst case, approximately $|V| + |E| + \text{overhead}$. $|V|$ accounts for the expanding of the core, at most all vertices in V, minus what are already in the core, would be included. $|E|$ accounts for calculating the in and out-degrees for the candidate vertices that are not in the core but in the neighborhood of the core. The overhead is caused by recalculating the in and out-degrees of neighboring vertices every time the *FindCore* is recursively called. The number of these vertices is dependent on the size of the community we are building and the connectivity of the community to the rest of the network, but not the overall size of the network. For biological networks, the graphs we deal with are mostly sparse and small world, therefore, the running time of our algorithm will be close to linear.

Algorithm 1 *CommBuilder*(G, s, f)
1: $G(V, E)$ is the input graph with vertex set V and edge set E.
2: s is the seed vertex, f is the affinity threshold.
3: $N \leftarrow \{\text{Adjacency list of } s\} \cup \{s\}$

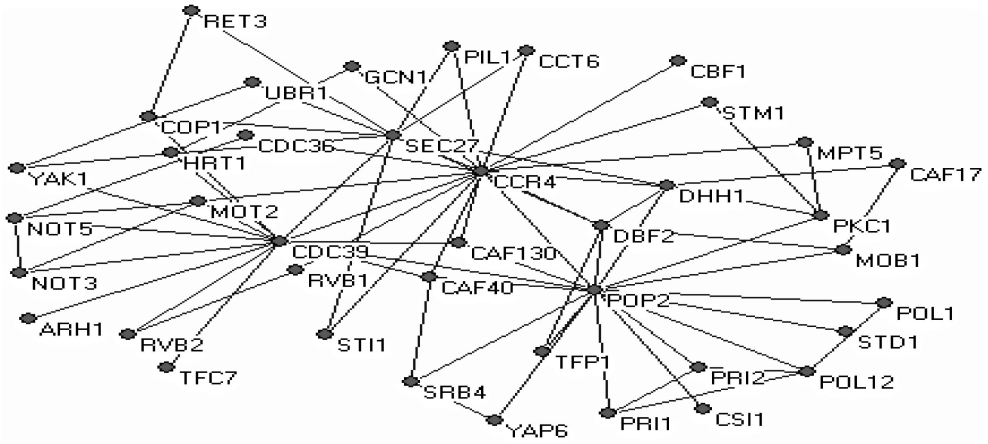


Fig. 3. The CCR4-NOT community.

4: $C \leftarrow \text{FindCore}(N)$
5: $C' \leftarrow \text{ExpandCore}(C, f)$
6: **return** C'

7: $\text{FindCore}(N)$
8: **for each** $v \in N$
9: calculate $k_v^{in}(N)$
10: **end for**
11: $K_{min} \leftarrow \min\{k_v^{in}(N), v \in N\}$
12: $K_{max} \leftarrow \max\{k_v^{in}(N), v \in N\}$
13: **if** $K_{min} = K_{max}$ **then return** N
14: **else return** $\text{FindCore}(N - \{v\}, k_v^{in}(N) = K_{min})$

15:$\text{ExpandCore}(C, f)$
16: $D \leftarrow \cup_{(v,w)\in E, v\in C, w\notin C}\{v, w\}$
17: $C' \leftarrow C$
18: **for each** $t \in D$ **and** $t \notin C$
19: calculate $k_t^{in}(D)$
20: calculate $k_t^{out}(D)$
21: **if** $k_t^{in}(D) > k_t^{out}(D)$ **or** $k_t^{in}(D)/|D| > f$ **then**
$C' \leftarrow C' \cup \{t\}$
22: **end for**
23: **if** $C' = C$ **then return** C
24: **else return** $\text{ExpandCore}(C', f)$

3.3.2 Experiment Results

To test our algorithm, we apply it to two biomolecular networks: 1) the yeast interaction network and 2) the chromatin protein interaction network.

Yeast Interaction Network. Because there is no alternative approach to our method, we decided to compare the performance of our algorithm to the work on predicting protein complex membership by Asthana et al. [1]. Asthana et al. reported the results of queries with four complexes using probabilistic network reliability (we will refer to their work as the PNR method in the following discussion). Four communities are identified by *CommBuilder* using one protein as a seed from each of the query complexes used by the PNR method. The seed protein is selected randomly from the "core" protein set. The figures for visualizing the identified communities are created using Pajek [4]. The community figures are extracted from the network we build using the above-mentioned data set with out-of-community connections omitted. The proteins in each community are annotated with a brief description obtained from the MIPS complex catalogue database. As a comparison, we use *Complexpander*, an implementation of the PNR method [1] and available at http://llama.med.harvard.edu/Software.html, to predict cocomplex using the core protein set that contains the same seed protein used by *CommBuilder*. For all of our queries when using *Complexpander*, we select the option of using the MIPS complex catalogue database. We record the ranking of the members in our identified communities that also appear in the cocomplex candidate list predicted by *Complexpander*. Two communities are discussed below due to space limitations.

The first community is discovered using NOT3 as the seed (Fig. 3). NOT3 is a known component protein of the CCR4-NOT complex, which is a global regulator of gene expression and involved in such functions as transcription regulation and DNA damage responses. The MIPS complex catalogue lists five proteins for NOT complex and 13 proteins (including the five NOT complex proteins) for the CCR4 complex. The NOT community identified is composed of 40 members, as shown in Table 5. All five NOT complex proteins listed in MIPS and 11 of the 13 CCR4 complex proteins are members of the community (the two missing CCR4 proteins may result from their low connectivity to the core). POL1, POL2, PRI1, and PRI2 are members of the DNA polymerase alpha (I)-primase complex, as listed in MIPS. RVB1, PIL1, UBR1, and STI1 have been grouped together with CCR4, CDC39, CDC36, and POP2 by systematic analysis [23]. The community also contains 20 out of 26 proteins of a complex that probably is involved in transcription and DNA/chromatin structure maintenance [16].

The second community is identified by using RFC2 as the seed (Fig. 4). RFC2 is a component of the RFC (replication factor C) complex, the "clamp loader," which plays an essential role in DNA replication and DNA repair. The community identified by our algorithm has 17 members. All five proteins of RFC complex listed in the MIPS complex catalogue database are members of this community, as shown in Table 6. All but one member in this community are in the functional category of DNA recombination and DNA repair or cell cycle checkpoints according to MIPS. This community also includes the top eight ranked proteins predicted by *Complexpander*.

TABLE 5
The CCR4-Not Community

Protein	*Alias*	*Description*	*Rank*
YDR376w	ARH1	mitochondrial protein putative ferredoxin-NADP+ reductase	38
YGR134w	CAF130†	CCR4 Associated Factor 130 kDa	8
YJR122w	CAF17*	CCR4 associated factor	
YNL288w	CAF40†	CCR4 Associated Factor 40 kDa	9
YJR060w	CBF1	centromere binding factor 1	
YAL021c	CCR4*†	transcriptional regulator	3
YDR188w	CCT6†	component of chaperonin-containing T-complex (zeta subunit)	30
YDL165w	CDC36*†	transcription factor	40
YCR093w	CDC39*†	nuclear protein	1
YDL145c	COP1†	coatomer complex alpha chain of secretory pathway vesicles	11
YMR025w	CSI1	Subunit of the Cop9 signalosome, involved in adaptation to pheromone signaling	46
YGR092w	DBF2*	ser/thr protein kinase related to Dbf20p	6
YDL160c	DHH1*	DExD/H-box helicase, stimulates mRNA decapping,	17
YGL195w	GCN1†	translational activator	26
YOL133w	HRT1	Skp1-Cullin-F-box ubiquitin protein ligase (SCF) subunit	
YIL106w	MOB1*	required for completion of mitosis and maintenance of ploidy	10
YER068w	MOT2*†	transcriptional repressor	2
YGL178w	MPT5	multicopy suppressor of POP2	
YIL038c	NOT3*†	general negative regulator of transcription, subunit 3	
YPR072w	NOT5*†	component of the NOT protein complex	5
YGR086c	PIL1	Long chain base-responsive inhibitor of protein kinases Phk1p and Phk2p, acts along with Lsp1p to down-regulate heat stress resistance	
YBL105c	PKC1	ser/thr protein kinase	
YNL102w	POL1†	DNA-directed DNA polymerase alpha, 180 KD subunit	32
YBL035c	POL12†	DNA-directed DNA polymerase alpha, 70 KD subunit	28
YNR052c	POP2*†	required for glucose derepression	4
YIR008c	PRI1†	DNA-directed DNA polymerase alpha 48kDa subunit (DNA primase)	34
YKL045w	PRI2†	DNA-directed DNA polymerase alpha , 58 KD subunit (DNA primase)	31
YPL010w	RET3	coatomer complex zeta chain	39
YDR190c	RVB1	RUVB-like protein	29
YPL235w	RVB2†	RUVB-like protein	21
YGL137w	SEC27†	coatomer complex beta^ chain (beta^-cop) of secretory pathway vesicles	7
YER022w	SRB4	DNA-directed RNA polymerase II holoenzyme and Kornberg^s mediator (SRB) subcomplex subunit	44
YOR047c	STD1	dosage-dependent modulator of glucose repression	
YOR027w	STI1	stress-induced protein	
YLR150w	STM1	specific affinity for guanine-rich quadruplex nucleic acids	
YOR110w	TFC7†	TFIIIC (transcription initiation factor) subunit, 55 kDa	25
YDL185w	TFP1†	encodes 3 region protein which is self-spliced into TFP1p and PI-SceI	27
YGR184c	UBR1	ubiquitin-protein ligase	
YJL141c	YAK1	ser/thr protein kinase	
YDR259c	YAP6	transcription factor, of a fungal-specific family of bzip proteins	

Proteins belonging to the CCR4-Not complex listed in MIPS are indicated by (*) and proteins considered to be involved in transcription and DNA/chromatin structure maintenance are indicated by (†).

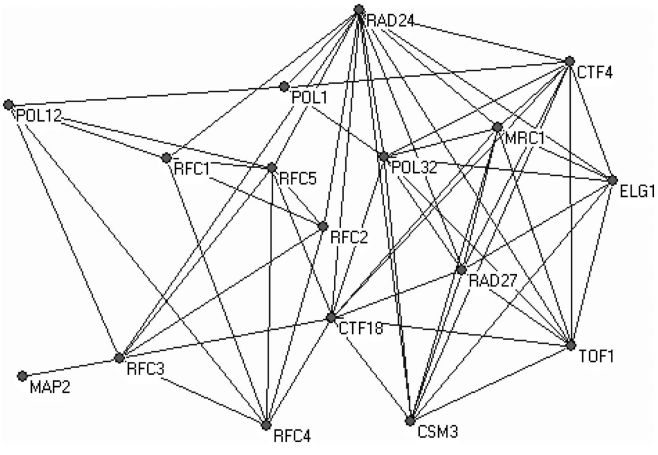


Fig. 4. The RFC community.

Chromatin Protein Interaction Network. The chromatin network forms a large scale-free network with 1,600 nodes (chromatin proteins) and 9,980 edges (to represent the protein-protein interaction of these proteins. Since there is no curated database related to the chromatin biological complexes identified by our methods. We rely on our domain expert Professor Lechner of the Biological Science and Technology Department at Drexel University to verify the biological meanings of those complexes. Most of the complexes have a nice agreement with the domain experts' knowledge. Fig. 5 shows two samples of the chromatin complexes.

4 Conclusion

In this paper, we present a novel unified method Bio-IEDM to extract biological knowledge from biomedical literature and identify potential biological meaningful communities from the derived biomolecular networks. Our method is

TABLE 6
The RFC Community

Protein	*Alias*	*Description*	*Rank*
YMR048w	CSM3†	Protein required for accurate chromosome segregation during meiosis	
YMR078c	CTF18†	required for accurate chromosome transmission in mitosis and maintenance of normal telomere length	6
YPR135w	CTF4†	DNA-directed DNA polymerase alpha-binding protein	
YOR144c	ELG1†	Protein required for S phase progression and telomere homeostasis, forms an alternative replication factor C complex important for DNA replication and genome integrity	7
YBL091c	MAP2	methionine aminopeptidase, isoform 2	
YCL061c	MRC1†	Mediator of the Replication Checkpoint	
YNL102w	POL1†	DNA-directed DNA polymerase alpha, 180 KD subunit	19
YBL035c	POL12†	DNA-directed DNA polymerase alpha, 70 KD subunit	5
YJR043c	POL32†	polymerase-associated gene, third (55 kDa) subunit of DNA polymerase delta	
YER173w	RAD24†	cell cycle checkpoint protein	1
YKL113c	RAD27†	ssDNA endonuclease and 5^-3^exonuclease	
YOR217w	RFC1*†	DNA replication factor C, 95 KD subunit	8
YJR068w	RFC2*†	DNA replication factor C, 41 KD subunit	
YNL290w	RFC3*†	DNA replication factor C, 40 kDa subunit	2
YOL094c	RFC4*†	DNA replication factor C, 37 kDa subunit	4
YBR087w	RFC5*†	DNA replication factor C, 40 KD subunit	3
YNL273w	TOF1†	topoisomerase I interacting factor 1	

Proteins belonging to the RFC complex listed in MIPS are indicated by (*) *and proteins listed in the functional category of DNA recombination and DNA repair or cell cycle checkpoints by MIPS are indicated by* (†).

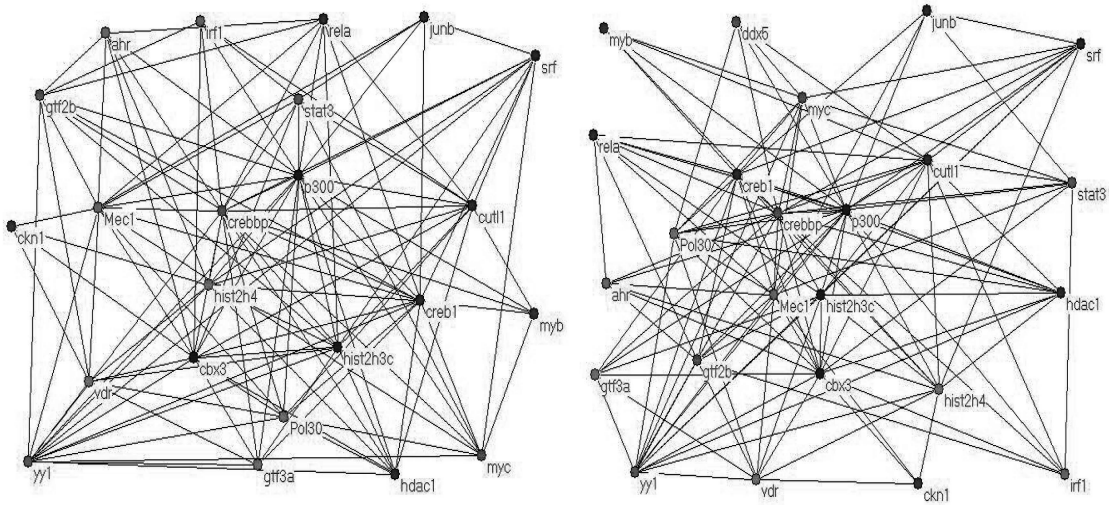


Fig. 5. Two chromatin complexes.

efficient enough to work in large online biomedical literature databases and flexible enough to be applied in very complicated domains with little human intervention. Our system, BIO-IEDM, can be used to extract many binary relationships such as protein-protein interaction, cell signaling, or protein-DNA interactions from a large collection of text files once the name dictionary of the studied object is provided and is a very useful tool for functional bioinformatics. The biological complexes will help uncover hidden relationships and complexes governing genomic operations. The contributions of our research approach are as follows:

- Automatic query generation for effective retrieval from large biomedical literature. We introduce a novel automatic query-based technique to identify Web documents that are promising for the extraction of relations from text while assuming only a minimal search interface to the biomedical literature databases. It automatically discovers the characteristics of documents that are useful for extraction of a target relation and generates queries in each iteration to select potentially useful documents from the text databases.
- Dual reinforcement information extraction for pattern generation and tuple extraction. The whole procedure is unsupervised, with no human intervention except for a few seed tuples provided by the user in the very beginning. Also, it introduces a strategy for evaluating the quality of the patterns and the tuples that are generated in each iteration of the extraction process. Only those tuples and patterns that are regarded as being "sufficiently reliable" will be kept by it for the following iteration of the system. These new strategies for generation and filtering of patterns and tuples improve the quality of the extracted tuples and patterns significantly.
- Our approach scales very well in huge collections in the biomedical literature databases because it does not need to scan every document. Since the only

domain-dependent component in our approach is the initial seed tuples, our system is easy to port to a new domain.
- Unlike other learning-based methods, which require parsing as the prerequisite in order to build a classification models, our approach works directly on the plain-text representation and needs much less manual intervention, without the laborious text preprocessing work.
- Our novel data mining algorithm provides an efficient way to detect a protein community from a seed. Experimental results have shown clear structural and functional relationships among members of the discovered community.

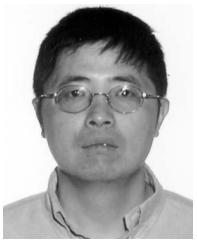
ACKNOWLEDGMENTS

The work is supported in part by research grants from the US National Science Foundation (NSF): NSF Career IIS 0448023, NSF CCF 0514679, and the Pennsylvania Department of Health (#240205, 240196). The authors thank Professor Lechner for providing the chromatin protein name list.

REFERENCES

[1] S. Asthana, O.D. King, F.D. Gibbons, and F.P. Roth, "Predicting Protein Complex Membership Using Probabilistic Network Reliability," *Genome Research,* vol. 14, pp. 1170-1175, 2004.
[2] G.D. Bader, I. Donaldson, C. Wolting, B.F. Quellette, T. Pawson, and C.W. Hogue, "BIND—The Biomolecular Interaction Network Database," *Nucleic Acids Research,* vol. 29, no. 1, pp. 242-245, 2001.
[3] A.-L. Barabási and A. Reka, "Emergence of Scaling in Random Networks," *Science,* vol. 286, pp. 509-512, Oct. 1999.
[4] V. Batagelj and A. Mrvar, "Pajek: Program for Large Network Analysis," *Connections,* vol. 21, pp. 47-57, 1998.
[5] C. Blaschke, M.A. Andrade, C. Ouzounis, and A. Valencia, "Automatic Extraction of Biological Information from Scientific Text: Protein-Protein Interactions," *Proc. Int'l Conf. Intelligent Systems in Molecular Biology,* pp. 60-67, 1999.
[6] A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Training," *Proc. 11th Ann. Conf. Computational Learning Theory,* pp. 92-100, 1998.
[7] S. Brin, "Extracting Patterns and Relations from the World Wide Web," *Proc. 1998 Int'l Workshop Web and Databases (WebDB '98),* pp. 172-183, 1998.
[8] B. Cestnik, "Estimating Probabilities: A Crucial Task in Machine Learning," *Proc. Europe Conf. AI,* 1990.
[9] J.H. Chiang and H.H. Yu, "MeKE; Discovering the Functions of Gene Products from Biomedical Literature via Sentence Alignment," *Bioinformatics,* vol. 19, no. 11, pp. 1417-1422, 2003.
[10] W. Cohen, "Fast Effective Rule Induction," *Proc. Int'l Conf. Machine Learning (ICML '95),* 1995.
[11] B. de Brujin and J. Martin, "Literature Mining in Molecular Biology," *Proc. EFMI Workshop Natural Language,* pp. 1-5, 2002.
[12] J. Ding, D. Berleant, D. Nettleton, and E. Wurtele, "Mining Medline: Abstracts, Sentences, or Phrases," *Proc. Pacific Symp. Biocomputing (PSB '02),* pp. 326-337, 2002.
[13] L. Donetti and M.A. Munoz, "Detecting Network Communities: A New Systematic and Efficient Algorithm," *J. Statistical Mechanics,* P10012, 2004.
[14] G.W. Flake, S.R. Lawrence, C.L. Giles, and F.M. Coetzee, "Self-Organization and Identification of Web Communities," *Computer,* vol. 35, pp. 66-71, 2002.
[15] K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi, "Toward Information Extraction: Identifying Protein Names from Biological Papers," *Proc. Pacific Symp. Biocomputing (PSB '98),* pp. 707-718, 1998.
[16] A.-C. Gavin et al., "Functional Organization of the Yeast Proteome by Systematic Analysis of Protein Complexes," *Nature,* vol. 415, pp. 141-147, 2002.
[17] M. Gavin and M.E.J. Newman, "Community Structure in Social and Biological Networks," *Proc. Nat'l Academy of Sciences USA,* vol. 99, pp. 7821-7826, 2002.
[18] U. Hahn, M. Romacker, and S. Schulz, "Creating Knowledge Repositories from Biomedical Reports: The MedSYnDikate Text Mining System," *Proc. Pacific Symp. Biocomputing (PSB '02),* pp. 338-349, 2002.
[19] T. Hasegawa, S. Sekine, and G. Ralph, "Discovering Relations among Named Entities from Large Corpora," *Proc. Ann. Meeting Assoc. Computational Linguistics (ACL '04),* 2004.
[20] L.H. Hartwell, J.J. Hopfield, S. Leibler, and A.W. Murray, "From Molecular to Modular Cell Biology," *Nature,* vol. 402, pp. C47-C52, 1999.
[21] R.F. Hashimoto, S. Kim, I. Shmulevich, W. Zhang, M.L. Bittner, and E.R. Dougherty, "Growing Genetic Regulatory Networks from Seed Genes," *Bioinformatics,* vol. 20, no. 8, pp. 1241-1247, 2004.
[22] L. Hirschman, J.C. Park, J. Tsujil, L. Wong, and C.H. Wu, "Accomplishments and Challenges in Literature Data Mining for Biology," *Bioinformatics,* vol. 18, no. 12, pp. 1553-1561, 2002.
[23] Y. Ho et al., "Systematic Identification of Protein Complexes in Saccharmyces Cerevisiae by Mass Spectrometry," *Nature,* vol. 415, pp. 180-183, 2002.
[24] P. Holme, M. Huss, and H. Jeong, "Subnetwork Hierarchies of Biochemical Pathways," *Bioinformatics,* vol. 19, no. 4, pp. 532-538, 2003.
[25] X. Hu and N. Cercone, "Discovering Maximal Generalized Decision Rules through Horizontal and Vertical Data Reduction," *Computational Intelligence: An Int'l J.,* vol. 17, no. 4, pp. 685-702, 2001.
[26] X. Hu, T.Y Lin, I.-Y. Song, X. Lin, I. Yoo, M. Lechner, and M. Song, "Ontology-Based Scalable and Portable Information Extraction System to Extract Biological Knowledge from Huge Collection of Biomedical Web Documents," *Proc. IEEE/ACM Web Intelligence Conf.,* pp. 77-83, Sept. 2004.
[27] X. Hu, I. Yoo, M. Song, J. Han, and M. Lechner, "Extracting and Mining Protein-Protein Interaction Network from Biomedical Literature," *Proc. IEEE Symp. Computational Intelligence in Bioinformatics and Computational Biology (IEEE CIBCB '04),* Oct. 2004.
[28] X. Hu, "Mining and Analyzing Scale-Free Protein-Protein Interaction Network," *Int'l J. Bioinformatics Research and Application,* vol. 1, no. 1, pp. 81-101, 2005.
[29] R. Jansen, N. Lan, J. Qian, and M. Gerstein, "Integration of Genomic Datasets to Predict Protein Complexes in Yeast," *J. Structural Functional Genomics,* vol. 2, pp. 71-81, 2002.
[30] M. Kanehisa and S. Goto, "A Systematic Analysis of Gene Functions by the Metabolic Pathway Database," *Theoretical and Computational Methods in Genome Research,* S. Suhai, ed., pp. 41-55, Plenum Press, 1997.
[31] J. Kleinberg, "Authoritative Sources in a Hyperlinked Environment," *Proc. Ninth ACM-SIAM Symp. Discrete Algorithms,* 1998.
[32] B. Liu, W. Hsu, and Y. Ma, "Integrating Classification and Association Rule Mining," *Proc. Int'l Conf. Knowledge Discovery and Dataming (KDD '98),* 1998.
[33] E.M. Marcott, I. Xenarios, and D. Eisenberg, "Mining Literature for Protein-Protein Interactions," *Bioinformatics,* vol. 17, no. 4, pp. 359-363, 2001.
[34] M.E.J. Newman, "The Structure and Function of Complex Networks," *SIAM Rev.,* vol. 45, no. 2, pp. 167-256, 2003.
[35] M.E.J. Newman, "Detecting Community Structure in Networks," *European Physics J. B,* vol. 38, pp. 321-330, 2004.
[36] T. Ono, H. Hishigaki, A. Tanigami, and T. Takagi, "Automated Extraction of Information on Protein-Protein Interactions from the Biological Literature," *Bioinformatics,* vol. 17, no. 2, pp. 155-161, 2001.
[37] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and Identifying Communities in Networks," *Proc. Nat'l Academy of Sciences USA,* vol. 101, pp. 2658-2663, 2004.
[38] E. Riloff, "Automatically Generating Extraction Patterns from Untagged Text," *Proc. 13th Nat'l Conf. Artificial Intelligenc (AAAI '96),* pp. 1044-1049, 1996.
[39] S. Soderland, "Learning Information Extraction Rules for Semi-Structured and Free Text," *Machine Learning,* vol. 34, pp. 233-272, 1999.
[40] V. Spirin and L.A. Mirny, "Protein Complexes and Functional Modules in Molecular Networks," *Proc. Nat'l Academy Sciences USA,* vol. 100, pp. 12123-12128, 2003.

[41] B.J. Stapley and G. Benoit, "Biobibliometrics: Information Retrieval and Visualization from Co-Occurrences of Gene Names in Medline Abstracts," *Proc. Pacific Symp. Biocomputing (PSB '00)*, pp. 529-540, 2000.
[42] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.
[43] J. Weizenbaum, "ELIZA—A Computer Program for the Study of Natural Language Communications between Men and Machine," *Comm. ACM*, vol. 9, pp. 36-45, 1966.
[44] S. White and P. Smyth, "A Spectral Clustering Approach to Finding Communities in Graphs," *Proc. SIAM Int'l Conf. Data Mining*, 2005.
[45] D. Wilkinson and B.A. Huberman, "A Method for Finding Communities of Related Genes," *Proc. Nat'l Academy of Sciences USA*, vol. 101, supplement 1, pp. 5241-5248, 2004.
[46] D. Wu and X. Hu, "An Efficient Approach to Detecting a Protein Community from a Seed," *Proc. IEEE Symp. Computational Intelligence in Bioinformatics and Computational Biology*, 2005.
[47] I. Xenarios, E. Fernandez, L. Salwinski, X.J. Duan, M.J. Thompson, E.M. Marcotte, and D. Eisenberg, "DIP: The Database of Interacting Proteins: 2001 Update," *Nucleic Acids Reg.*, vol. 29, pp. 239-241, 2001.
[48] D. Zelenko, C. Aone, and A. Richardella, "Kernel Methods for Relation Extraction," *J. Machine Learning Research*, vol. 3, pp. 1083-1106, 2003.



Xiaohua (Tony) Hu is currently an assistant professor in computer science in the College of Information Science and Technology, Drexel University, Philadelphia, Pennsylvania. His current research interests are biomedical literature data mining, bioinformatics, text mining, and rough sets. He has been published more than 120 peer-reviewed publications in various journals, conferences, and books and served as a conference chair, program chair and program committee member for more than 50 international conferences in the above areas. He has received a few prestigious awards, including the 2005 US National Science Foundation Career award, the best paper award at the 2004 IEEE CIBCB, and the 2001 IEEE Data Mining Outstanding Service Award. He is the founding editor-in-chief of the *International Journal of Data Mining and Bioinformatics*. He is a member of the IEEE.

Daniel D. Wu received the BS degree in biochemistry from Xiamen University in China, the MS degree in physiology (1996), and the MS degree in computer science (2001) from Pennsylvania State University. He is currently pursuing the PhD degree in the College of Information Science and Technology at Drexel University, Philadelphia, Pennsylvania. His research interests are in data mining, bioinformatics, and biomolecular network analysis.