Compositional Mining of Multirelational Biological Datasets

YING JIN, T. M. MURALI, and NAREN RAMAKRISHNAN
Virginia Tech

High-throughput biological screens are yielding ever-growing streams of information about multiple aspects of cellular activity. As more and more categories of datasets come online, there is a corresponding multitude of ways in which inferences can be chained across them, motivating the need for compositional data mining algorithms. In this article, we argue that such compositional data mining can be effectively realized by functionally cascading redescription mining and biclustering algorithms as primitives. Both these primitives mirror shifts of vocabulary that can be composed in arbitrary ways to create rich chains of inferences. Given a relational database and its schema, we show how the schema can be automatically compiled into a compositional data mining program, and how different domains in the schema can be related through logical sequences of biclustering and redescription invocations. This feature allows us to rapidly prototype new data mining applications, yielding greater understanding of scientific datasets. We describe two applications of compositional data mining: (i) matching terms across categories of the Gene Ontology and (ii) understanding the molecular mechanisms underlying stress response in human cells.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—*Data mining*; I.2.6 [**Artificial Intelligence**]: Learning

General Terms: Algorithms

Additional Key Words and Phrases: Biclustering, bioinformatics, compositional data mining, inductive logic programming, redescription mining

ACM Reference Format:

Jin, Y., Murali, T. M. and Ramakrishnan. N. 2008. Compositional mining of multirelational biological datasets. ACM Trans. Knowl. Discov. Data. 2, 1, Article 2 (March 2008), 35 pages. DOI = 10.1145/1342320.1342322 http://doi.acm.org/10.1145/1342320.1342322

1. INTRODUCTION

Our ability to interrogate the cell and computationally assimilate its answers is improving at a dramatic pace. For instance, the study of even a focused aspect of cellular activity, such as gene action, now benefits from multiple high-throughput data acquisition technologies such as microarrays [Ball et al. 2005],

Author's address: N. Ramakrishnan; email: Naren@cs.vt.edu.

genome-wide deletion screens [Carpenter and Sabatini 2004], and RNAi assays [Gunsalus and Piano 2005; Matzke and Birchler 2005; Matzke and Matzke 2004]. As more and more categories of biological data come online, there is a corresponding multitude of ways in which inferences can be chained across them, making it infeasible to prototype software for every conceivable analysis methodology. Different biologists have different needs and perspectives, and it is difficult to anticipate all the ways in which computational pipelines can be organized.

Consider the following two scenarios from bioinformatics applications. In the first, Scientist A desires to identify a small set of *C. elegans* genes (perhaps encoding transcription factors) to knock down (via RNAi) in order to confer improved desiccation tolerance in the nematode. Scientist A might begin by identifying those genes whose knock-down produces phenotypes related to improved desiccation tolerance and then find one or more transcription factors that combinatorially control the expression of these genes. In the second scenario, Scientist B is interested in analyzing similarities across gene expression programs underlying aging in *C. elegans* and *D. melanogaster*. Scientist B might use DNA microarrays to measure gene expression across a wide time span in aging worms and flies; analyze these datasets individually to find clusters of genes that are coexpressed under a subset of the time points; and determine if genes in a *C. elegans* cluster have a significant number of orthologs in a *D. melanogaster* cluster. To support such arbitrary lines of reasoning, we need novel software tools that allow biologists to uniformly decompose complex analytical functions in terms of primitives that reason about and relate entities across biological domains.

We argue for *compositional data mining* (CDM), which, as the name indicates, is a way to construct complex data mining functions from simpler data mining primitives. Key to this idea is focusing on small set of primitives that are powerful algorithms in their own right but which can be functionally cascaded in arbitrary ways. We present a software system (Proteus) that embodies the CDM concept using two such primitives—*redescriptions* and *biclusters*. These primitives serve complementary purposes and mirror shifts of vocabulary that often accompany logical chains of reasoning (e.g., transcription factors → regulated genes → knock-down phenotypes for the desiccation scenario; worm age → *C. elegans* genes → *D. melanogaster* orthologs → fly age in the aging scenario.) In our prior work [Murali and Kasif 2003; Parida and Ramakrishnan 2005; Pati et al. 2006; Ramakrishnan et al. 2004; Zaki and Ramakrishnan 2005], we have applied these primitives, individually, to gain significant insight into massive datasets. Using CDM, we combine their expressiveness to form chains of reasoning across domains.

The rest of this article is organized as follows. Section 2 uses examples to introduce the basic concepts underlying compositional data mining. Section 3 develops formalisms that capture the various elements of CDM. Section 4 presents various algorithms that together help mine compositional patterns. Experimental results are presented next, first showcasing the effectiveness of our algorithms and optimizations in Section 5, followed by, in Section 6, examples of knowledge discovered from two application case studies: matching terms

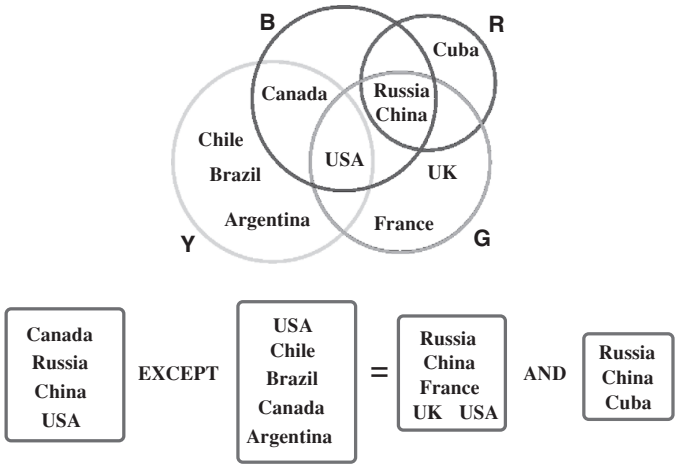


Fig. 1. (top) Example input to redescription mining. (bottom) Sample redescription. The expression $B - Y$ can be redescribed into $G \cap R$.

across categories of the gene ontology (GO) and understanding the molecular mechanisms underlying stress response in human cells. Related research and conclusions are presented finally, in Sections 7 and 8.

2. COMPOSITIONAL DATA MINING

Compositional data mining is not intended to be a one-size-fits-all data mining technique; rather, it is a way of problem decomposition based on the notions of biclusters and redescriptions. We begin by reviewing these primitives: whereas redescriptions relate object sets within a domain, biclusters relate object sets across domains.

2.1 Redescription Mining

As the term indicates, to redescribe something is to describe anew or to express the same concept in a different way. The input to redescription mining is a set of objects and a collection of subsets defined over this set. It is easiest to illustrate redescription mining using an everyday example. Consider the set of ten countries shown in Figure 1 and its four subsets, each of which denotes a meaningful grouping of countries according to some intensional definition. For instance, the colors (G) green, (R) red, (B) blue, and (Y) yellow (from right, counterclockwise) refer to the sets "permanent members of the UN security council," "countries with a history of communism," "countries with land area $> 3,000,000$ square miles," and "popular tourist destinations in the Americas (North and South)." We will refer to such sets as *descriptors*. A redescription is a shift of vocabulary and the goal of redescription mining is to identify subsets that can be defined in at least two ways using the given descriptors. An example redescription for this dataset is "Countries with land area $> 3,000,000$ square miles outside of the Americas" are the same as "Permanent members of the UN security council who have a history of communism." This redescription defines

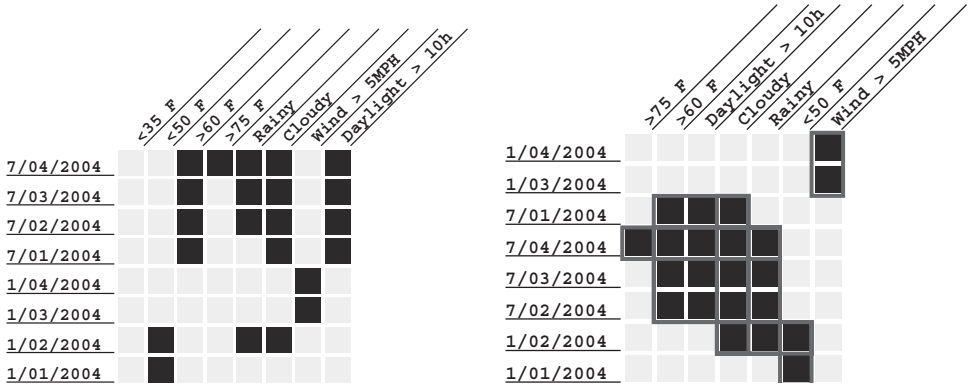


Fig. 2. (left) Example input to biclustering. (right) Layout of computed biclusters.

the set {Russia, China}, first by a set intersection of political indicators ($G \cap R$), and second by a set difference involving geographical descriptors ($B - Y$). Notice that neither the set of objects to be redescribed nor the ways in which descriptor expressions should be constructed is input to the algorithm. The underlying premise of redescription analysis is that sets that can indeed be defined in (at least) two ways are likely to exhibit concerted behavior and are, hence, interesting.

2.2 Biclustering

The input to bicluster mining [Madeira and Oliveira 2004] is a set of instances of a relationship between two or more domains. Figure 2 describes relationships between dates (rows) and weather conditions (columns) in Blacksburg, VA. A bicluster is a subset of rows along with a subset of columns with the property that each row element is related to each column element (later we will utilize stricter notions of biclusters, but this definition will suffice for this example). Figure 2 (right) lays out the seven biclusters in the matrix as contiguous submatrices by reordering the rows and columns of the matrix [Grothaus et al. 2006], repeating rows and columns if necessary. For example, the bicluster spanning rows three through six and columns two through four states that each of the four days from July 1 to 4, 2004 experienced each of the weather conditions "> 60 F," "Daylight > 10 h," and "Cloudy."

2.3 Composing Biclusters and Redescriptions

Both redescriptions and biclusters have direct applications in bioinformatics. Redescriptions are useful in relating gene sets from vocabularies based on cellular location (e.g., "genes localized in the mitochondrion"), transcriptional activity (e.g., "genes up-regulated two-fold or more in heat stress"), protein function (e.g., "genes encoding proteins that form the Immunoglobin complex"), or biological pathway involvement (e.g., "genes involved in glucose biosynthesis"). Similarly, biclusters are useful when we want to identify, for example, sets of genes together with sets of experiments or sets of phenotypes that exhibit

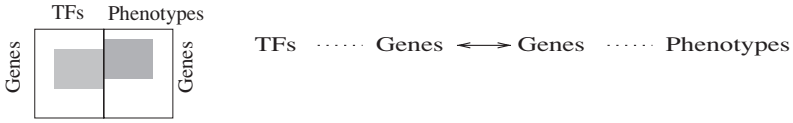


Fig. 3. Finding transcription factors (TFs) whose knock-down induces improved desiccation tolerance in *C. elegans*. (left) Two biclusters (shaded rectangles) joined at the gene interface using an (approximate) redescription. (right) Compositional data mining schema, displaying the sequence of primitives. Here, arrows indicate redescriptions, and dotted lines indicate biclusters.

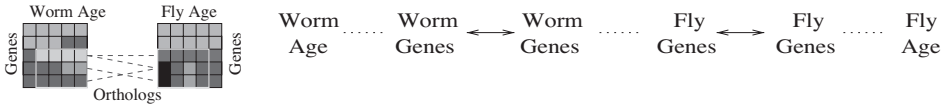


Fig. 4. Finding shared gene expression programs in adult aging in *C. elegans* and *D. melanogaster*. (left) Three biclusters with redescription mining at the two gene interfaces. (right) Compositional data mining schema, displaying the sequence of primitives. As before, arrows indicate redescriptions, and dotted lines indicate biclusterings.

concerted co-occurrences. However, they have complementary advantages and limitations.

Redescriptions not only identify concerted sets but can also give meaningful characterizations of them in terms of data descriptors. This capability is akin to conceptual clustering [Fisher 1987; Michalski 1980], where clusters are required to satisfy describability constraints. On the other hand, biclusters extensionally enumerate elements of subsets from both domains; we must do a post-analysis of the contents of these sets to describe them. Conversely, redescription mining requires that all descriptors be stated over a common universal set, so that data spanning multiple relations must be collapsed into one of the underlying domains. For instance, a relationship between genes and transcription factors might be used to define descriptors over genes. On the other hand, biclustering retains the relational nature of information and models patterns in relations. It is hence natural to combine their complementary capabilities.

To illustrate CDM, let us revisit the two scenarios from the introduction. The first scenario can be modeled by mining biclusters between genes and the transcription factors that regulate them, mining biclusters between genes and the phenotypes that result when they are knocked down, and connecting one side of the first bicluster to one side of the second bicluster using a redescription (see Figure 3). The second scenario can be modeled by mining three biclusters—for the relationship between worm genes and worm age, for the relationship between fly genes and fly age, and for the orthology relationship between fly genes and worm genes (see Figure 4). To cascade these three biclusters together, we can use two redescriptions as intermediaries, one redescribing worm genes, and the other redescribing fly genes. We can think of such cascading as either the biclustering algorithm supplying descriptors to the redescription algorithm, or the redescription algorithm specifying the objects that must participate in the biclustering. The results of such compositions can be read sequentially from one

end to the other, not unlike a story. For instance, for the first scenario above, we might find that "genes regulated by superoxide dismutase and catalase transcription factors, when knocked down, will result in cells with a phenotype of hypersensitivity to oxidative stress." In general, such compositions can induce a graph of arbitrary topology in the underlying data model, as we will see later.

Unlike the example in Figure 1, observe that both the CDM scenarios from Figures 3 and 4 do not involve any constructive induction of descriptors in the redescriptions. There are situations where this feature is important, for example, we may desire to find patterns such as "genes regulated by superoxide dismutase and catalase transcription factors *but not* by transcription factors that control the cell cycle, when knocked down, will result in cells with a phenotype of hypersensitivity to oxidative stress *as well as* abnormal cell size." To mine such patterns, each redescription must potentially relate two or more biclusters on either side. In this first paper on CDM, we define descriptors as the "projections" of biclusters onto the relevant domains and focus on redescriptions with only one bicluster on each side, rather than on connecting set-theoretic combination of bicluster projections.

The Proteus vision of a CDM system is that a biologist can merely specify the domains that must participate in the composition (e.g., "TFs" and "phenotypes") and the system automatically identifies a suitable composition of mining algorithms to relate the given domains. Observe that it can be infeasible to realize CDM by propositionalization, that is, by first "multiplying" out the original multirelational dataset into a single-relation dataset, mining patterns in the integrated set, and then unpacking the pattern to relate the given domains. Although propositionalization has proved to be viable in traditional inductive logic programming [Lavrac and Flach 2001], such algorithms only need to relate individual *objects* across domains, whereas we must relate *sets* across domains, which are much larger in number and not defined *a priori*. In essence, CDM is relational knowledge discovery [Dzeroski and Lavrac 2001] over sets, instead of objects. It is also wasteful to organize independent redescription and biclustering results across the different domains and relationships, since many of the patterns mined would not participate in any connections.

Another approach to CDM might be to start by computing biclusters in one relationship and use them to constrain the mining [Bayardo 2002] of biclusters in a neighboring relationship. However, such constraint-based mining is ill equipped to deal with the arbitrary expansion and contraction of descriptor sizes that CDM must support. Nevertheless, there are several significant structural properties of CDM patterns that we will exploit to design efficient mining algorithms.

The key contributions of this article are as follows:

(1) We formulate the notion of compositional data mining as an approach to better conceptualize structured data mining problems. Rather than developing special purpose algorithms for every new type of dataset or analysis goal, CDM helps to organize knowledge discovery tasks in a modular manner.
(2) Since CDM patterns connect sets of entities through alternating biclusters and redescriptions, we present a new "compose then compute" algorithm

that combines two biclustering and one redescription mining invocations in a single step. This primitive significantly speeds up the composition process and also avoids wasteful data mining.

(3) Using the pattern mined by this integrated algorithm as a primitive, we show how mining compositional patterns reduces to systematic searches for joins over a suitably defined "CDM schema." We can derive the CDM schema automatically from the original schema. Entities in the CDM schema represent *sets* of objects in the original schema. Recall that these sets are not defined *a priori*. They are mined by the compose then compute algorithm.

(4) We leverage classical levelwise principles, in the spirit of *Apriori* [Agrawal and Srikant 1994] and WARMR [Dehaspe and Toivonen 1999], and extend them to find CDM patterns. This extension greatly broadens the applicability of the optimizations in these algorithms, just as the query flocks paradigm [Tsur et al. 1998] generalized the *Apriori* "trick" to general conjunctive queries.

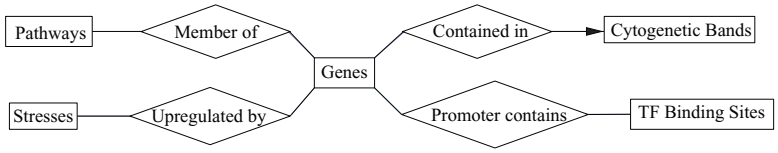
3. FORMALISMS

In this section, we introduce a sequence of formalisms beginning with database schemas, followed by data descriptors, redescriptions, and biclusters, culminating in CDM queries that will be of interest in this work. We use two running examples to illustrate these ideas. The first example relates four aspects of a gene's function and regulation: the pathways it is a member of, the (unique) cytogenetic band it is contained in, the transcription factor (TF) binding sites present in its promoter, and stresses that up-regulate the gene. The second example relates small molecules to diseases they may treat and to genes they up-regulate, and pathways to diseases they are implicated in and genes that are their members. We will refer to these examples as "Gene properties" and "Small molecules," respectively.

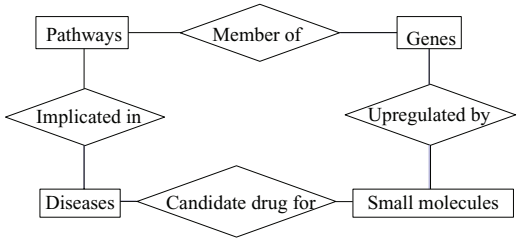
3.1 Database Schemas

An *entity set* is a set of objects from a particular domain, for example, genes, proteins, TF binding sites, or pathways. Objects in an entity set E can have values for a set of *properties*, denoted P_E. Given two entity sets E and F, a *(binary) relationship* $R(E, F)$ between E and F is a subset of $E \times F$; we say that R is *connected* to E and F. It is useful to view R both as a binary matrix and as a bipartite graph. For example, relationships may connect proteins to each other via physical interactions, genes to TF binding sites in their promotors, or genes to pathways they belong to. In this paper, we consider only binary relationships although relationships of higher cardinality can be re-stated in terms of (multiple) binary relationships.

Given a set $\mathcal{E}$ of entity sets and a set $\mathcal{R}$ of relationships between entity sets in $\mathcal{E}$, a *database schema* $\mathcal{S}(\mathcal{E}, \mathcal{R})$ is a connected bipartite graph whose node set is given by $\mathcal{E} \cup \mathcal{R}$ (i.e., includes both entity sets and relationships) and whose edge set comprises edges each of which connects a relationship in $\mathcal{R}$ to an entity set in $\mathcal{E}$. Observe that all nodes in $\mathcal{R}$ are constrained to have degree two in $\mathcal{S}$



(a) "Gene properties": database schema



(b) "Small molecules": database schema

Fig. 5. Database schemas for two examples.

whereas there are no degree constraints on the nodes in $\mathcal{E}$. Figure 5 displays the schema for our two examples.

Although typical database schema specification languages such as SQL DDLs capture more information, we use the term database schemas in this paper to primarily refer to the graph structure of entities and relationships.

3.2 Descriptors and Redescriptions

A *descriptor* over an entity set E identifies a subset of entities from E. The typical way to define a descriptor is as a Boolean expression over a subset of properties $Q \subseteq P_E$. For instance, the set of entities with a particular value for an attribute, for example, "the set of proteins with molecular weight equal to 100 kDa," is a descriptor. Relationships can also yield descriptors. For instance, using the relationship connecting genes to pathways they participate in, "genes in the Kit receptor pathway" constitutes a descriptor over genes. To accommodate such descriptors, it is useful to think of the set of properties P_E as being augmented from attribute-value definitions to relational definitions. Henceforth, we will use P_E to denote properties defined using both means. Given a descriptor d, we will denote the set of entities for which d is true by $E(d)$.

Two descriptors d_1 and d_2 over an entity set E are said to be *redescriptions* of each other, denoted $d_1 \Leftrightarrow d_2$, if they are distinct and approximately induce the same subset of entities from E. The distinctness condition rules out tautologies, for example, an equivalence such as $P_1 \cap P_2 \Leftrightarrow P_1 - (P_1 - P_2)$ is not interesting because it holds in *all* datasets. The second condition can be evaluated by measures such as the support and Jaccard's coefficient. The support of a redescription $d \Leftrightarrow d'$ is given by the cardinality of the intersection of both descriptors, that is, $|E(d) \cap E(d')|$. The Jaccard's coefficient of $d \Leftrightarrow d'$ is given by $\frac{|E(d) \cap E(d')|}{|E(d) \cup E(d')|}$. It is zero if the descriptors are disjoint and one if they are the same. We will typically use the support constraint as a parameter to redescription

mining and the Jaccard's coefficient (and other measures) to evaluate a mined redescription. We do so because biologists find it more natural to input the number of, say, common genes, rather than the Jaccard's coefficient.

We define the predicate $\rho(d, d')$ that is true if and only if the redescription $d \Leftrightarrow d'$ holds (at some support or Jaccard's coefficient level, which will be implicit in the context). Note that redescriptions are symmetric, that is, $\rho(d, d') \equiv \rho(d', d)$. We will sometimes abuse notation and use the expression $\rho(d, d')$ to refer to the redescription itself.

3.3 Biclusters

Let $R(E, F)$ be a relationship between entity sets E and F. A *bicluster* (E', F') *on* R is a set $E' \subseteq E$ and a set $F' \subseteq F$ such that $E' \times F' \subseteq R$, that is, every pair of entities in $E' \times F'$ belongs to R. Further, the bicluster (E', F') is *closed* if

(i) for every entity $e \in E - E'$, there is some entity $f \in F'$ such that $(e, f) \notin R$, and
(ii) for every entity $f \in F - F'$, there is some entity $e \in E'$ such that $(e, f) \notin R$.

That is, adding an entity in $E - E'$ or $F - F'$ to the bicluster will violate the condition defining the bicluster. We say that E' and F' are *projections* of the bicluster onto E and F, respectively. Observe that projections are a natural way to define descriptors over E and over F.

Similar to the redescription predicate ρ, we define a predicate $\beta(d, d')$ that is true if and only if descriptors d and d' constitute the projections of a closed bicluster. Observe that there is no requirement that d and d' be defined over the same entity set. Moreover, unlike redescriptions, except in special cases, $\beta(d, d')$ does not imply $\beta(d', d)$. To avoid confusion, we will present the arguments for β in the same order as the relationship from which it was derived. We will also use the expression $\beta(d, d')$ to refer to the closed bicluster (d, d').

We will find it convenient to expand a bicluster into a closed one. Given a bicluster (E', F'), its *closure* is any closed bicluster (E'', F'') such that $E' \subseteq E''$ and $F' \subseteq F''$. Note that unlike the notion of closures used in association rule mining [Zaki and Hsiao 2002], this definition allows multiple biclusters to be closures of a given bicluster. This aspect will become relevant when we present our algorithms for compositional data mining.

We note that if R is a one-to-one relationship from E to F, then every bicluster on R contains exactly one element from E and one element from F and the number of such biclusters is $|R|$. Furthermore, if R is many-to-one from E to F, then each bicluster on R contains exactly one element from F and the number of these biclusters is at most $|F|$. For many-many relationships, biclusters correspond to bicliques in the bipartite graph representing R.

In general, relationships can themselves have properties. For instance, gene expression data is a relationship between genes and samples, where each (gene, sample) pair is associated with an expression value. For such relationships, we will assume the existence of appropriate algorithms [Madeira and Oliveira 2004; Tanay et al. 2005] for biclustering numerical data (see Section 6.2 for an example).

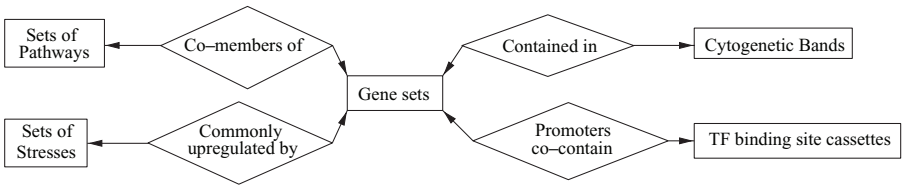


Fig. 6. "Gene properties": CDM schema.

As in the case of redescriptions, we will typically mine biclusters by imposing a minimum support constraint (which can be specified over either or both domains involved in the relationship).

3.4 CDM Schemas

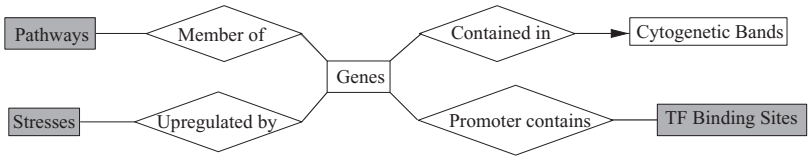
Given a database schema $\mathcal{S}(\mathcal{E}, \mathcal{R})$, its *CDM schema* $\mathcal{S}^\wedge(\mathcal{E}^\wedge, \mathcal{R}^\wedge)$ is another database schema whose entity sets and relationships have a one-to-one correspondence with the entity sets and relationships of $\mathcal{S}$ with the following properties:

(i) Every entity set E in $\mathcal{E}$ is mapped to another entity set $E^\wedge$ in $\mathcal{E}^\wedge$; each element of $E^\wedge$ is a subset of E.
(ii) Every relationship $R(E, F)$ in $\mathcal{R}$ is mapped to a relationship $R^\wedge(E^\wedge, F^\wedge)$ in $\mathcal{R}^\wedge$ between the entity sets $E^\wedge$ and $F^\wedge$.
(iii) If $(E', F') \in R^\wedge(E^\wedge, F^\wedge)$, then $\beta(E', F')$ is true in R, E' is an entity in $E^\wedge$, and F' is an entity in $F^\wedge$.

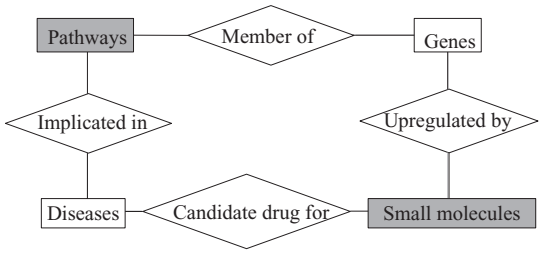
Thus, an entity in $\mathcal{S}^\wedge$ maps to a set of entities in $\mathcal{S}$. Figure 6 displays the CDM schema for the example in Figure 5(a): the entity set "Genes" is mapped to "Gene sets," the entity set "Stresses" is mapped to "Sets of stresses," and so on. Similarly, the members of a pair belonging to the "Co-member" relationship in $\mathcal{S}^\wedge$ are the projections, onto the "Pathways" and "Genes" entity sets, of a closed bicluster on the "Member of" relationship. Since the relationship "Contained in" is many-one from "Genes" to "Cytogenetic bands," the entity set "Cytogenetic bands" in the CDM schema represents single bands and not sets of them. Observe that redescriptions do not play a role in the CDM schema. (We will use them below in answering CDM queries.) Finally, the third condition in the formulation of the CDM schema implicitly enforces referential integrity constraints over the sets participating in all instances of relationships in $\mathcal{S}^\wedge$.

LEMMA 3.1. *If $R(E, F)$ is a relationship in $\mathcal{E}$, then $R^\wedge(E^\wedge, F^\wedge)$ is a one-to-one relationship.*

PROOF. Suppose that $R^\wedge(E^\wedge, F^\wedge)$ is not a one-to-one relationship and that two pairs (E', F') and (E', F'') belong to $R^\wedge(E^\wedge, F^\wedge)$, where $E' \in E$ and $F', F'' \in F$ and $F' \neq F''$. By definition of the CDM schema, both $\beta(E', F')$ and $\beta(E', F'')$ are true in R. Then $\beta(E', F' \cup F'')$ is also true; that is, the bicluster formed by E' and $F' \cup F''$ is also closed. Since $F' \neq F''$, both F' and F'' are contained in $F' \cup F''$, which violates the assumption that the original biclusters are closed. Therefore, $R^\wedge(E^\wedge, F^\wedge)$ is a one-to-one relationship. □



(a) "Gene properties": Database schema highlighting three entity sets in the sample query.



(b) "Small molecules": database schema highlighting the two entity sets in the sample query.

Fig. 7. Two example CDM queries posed over database schemas.

Observe that Lemma 3.1 holds irrespective of the nature of the relationship in R.

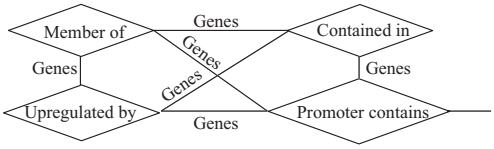
There may not be a natural notion of a closed bicluster for relationships that have numeric attributes. In such cases, we will construct biclusters that ensure that Lemma 3.1 still holds.

With the construction of the CDM schema, observe that we are able to connect *sets* of entities to each other via biclusters and redescriptions. The advantage of the above formulation is that a compositional mining query over the original schema $\mathcal{S}$ now reduces to a simple database join over the CDM schema $\mathcal{S}^\wedge$. In particular, optimizations such as query flocks [Tsur et al. 1998] can be readily applied to yield patterns that are actually comprised of sets of objects.

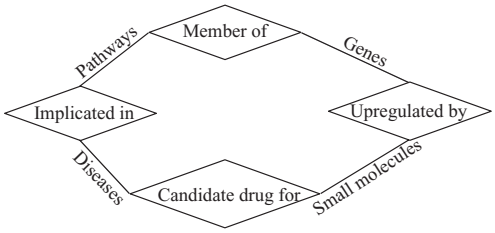
3.5 CDM Queries and Compositions

We now define the primary component of CDM queries and their results. A *CDM query* is a k-tuple $Q(E_1, E_2, \ldots, E_k)$, where $k \geq 2$ is an integer, $E_i \in \mathcal{E}$, $1 \leq i \leq k$, and the E_i's are distinct. Figure 7 illustrates two CDM queries, one for each of our examples. The first query specifies three entity sets: "Pathways," "Stresses," and TF Binding Sites. The second query specifies the entity sets "Pathways" and "Small molecules."

Informally, the semantics of the query is that the user is interested in compositions of biclusters and redescriptions involving the given entity sets, that is, all the specified k entity sets *must* participate in the composition. Note that the user specifies the CDM query in the context of the original schema $\mathcal{S}(\mathcal{E}, \mathcal{R})$ and that this formulation only specifies the entity sets she desires to participate



(a) "Gene properties": the relationship graph.



(b) "Small molecules": the relationship graph.

Fig. 8. Relationship graphs for the two illustrative CDM scenarios.

in the result. The user need not specify which relationships must participate in the query, or which other intermediate entity sets must be involved in the composition, since she may not know beforehand the intermediaries that will most usefully connect the entity sets of interest.

Observe that the user can obtain a trivial answer to such a CDM query by joining appropriate tables of the original schema. However, such answer will only yield compositions involving individual entities. As stated earlier, the crux of CDM is to compute compositions involving sets of entities.

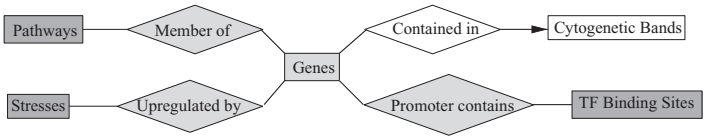
The precise interpretation of the CDM query can refer to computing all compositions, testing for the existence of (at least) a composition, or counting the number of compositions. In this paper, we develop the CDM methodology in the context of computing all compositions. (Algorithms other than those proposed here might be more suited when we are trying to answer existence or counting queries.) We will also show how to impose constraints similar to the minimum support constraint popular in association rule mining.

First, we define a transformation of the database schema $\mathcal{S}$ that we will use to translate CDM queries into composition plans. The *relationship graph* $\Gamma(\mathcal{S})$ *of a database schema* $\mathcal{S}$ is a graph such that

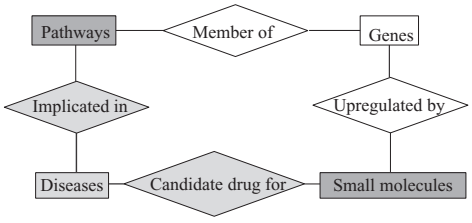
(1) nodes in $\Gamma(\mathcal{S})$ have a one-to-one correspondence with the relationships of $\mathcal{S}$,
(2) two nodes in $\Gamma(\mathcal{S})$ are connected by an edge if the corresponding relationships share a common entity set in $\mathcal{S}$. The edge is labeled by this common entity set.

Note that this concept is similar to the "relationship summary network" in Long et al. [2006] but captures the schema, instead of the instances. Informally, nodes in the relationship graph correspond to biclusters and edges correspond to redescriptions over the entity sets labeling the edges. Figure 8 illustrates the relationship graphs for our two examples.

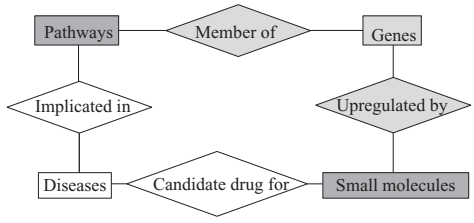
Given a CDM query $Q(E_1, E_2, \ldots, E_k)$ on the schema $\mathcal{S}(\mathcal{E}, \mathcal{R})$, we say that a subgraph $\mathcal{T}$ of $\mathcal{S}$ *matches* Q if $\mathcal{T}$ is connected and E_i is a node of $\mathcal{T}$, for every



(a) "Gene properties": subgraph matching the CDM query.



(b) "Small molecules": the first subgraph matching the CDM query.



(c) "Small molecules": the second subgraph matching the CDM query.

Fig. 9. Subgraphs matching CDM queries.

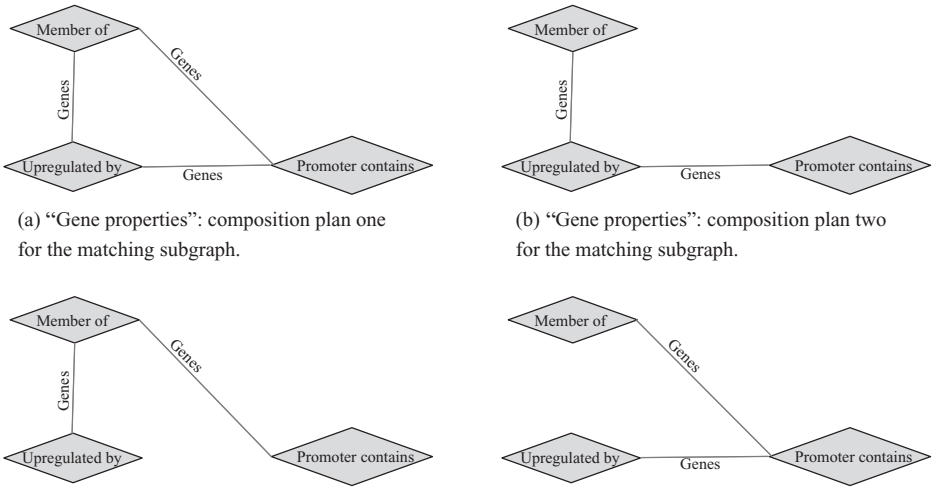
$1 \leq i \leq k$. Such a subgraph "fleshes" out the query by adding relationships and other entity sets in order to connect all the entity sets in the query. At this stage, we do not impose any constraints on the minimality of the subgraph that a query matches. Figure 9 displays the subgraphs matching the queries from Figure 7. Note that *two* subgraphs match the query for the "Small molecules" example. Moreover, the given schema for each of these examples is trivially a matching subgraph, which we do not display.

Now we define how to transform such a subgraph $\mathcal{T}$ into a subgraph of $\Gamma(\mathcal{S})$. Given a subgraph $\mathcal{T}$ of $\mathcal{S}$ that matches a query Q, the *relationship graph* $\Gamma(\mathcal{T})$ *of* $\mathcal{T}$ is the subgraph of $\Gamma(\mathcal{S})$ induced by the nodes that correspond to the relationships in $\mathcal{T}$. We also say that $\Gamma(\mathcal{T})$ *matches* the query Q. We observe without proof that $\Gamma(\mathcal{T})$ is unique and connected.

Next, we map relationship graphs matching a given CDM query to specific composition plans. Before we present the details of composition plans, it is helpful to have some additional definitions. We say that a closed bicluster $\beta(E', F')$ and a redescription $\rho(X, Y)$ *compose* if $F' = X$. We denote the composition by $\beta\rho(E', F', Y)$. Another way in which closed bicluster $\beta(E', F')$ and redescription $\rho(X, Y)$ may compose is if $E' = Y$, denoted by $\rho\beta(X, E', F')$. Similarly, we can achieve a composition involving two biclusters by introducing a suitable redescription in between: the composition $\beta\rho\beta(E', F', G', H')$ holds if $\beta(E', F')$, $\beta(G', H')$, and $\rho(F', G')$ together hold. Observe that the two biclusters in $\beta\rho\beta(E', F', G', H')$ could potentially be derived from different relationships although the types of F' and G' must be the same (for the redescription to hold). We use the $\beta\rho\beta$ predicates as building blocks for CDM.

Although not studied here in detail, we can also allow two redescriptions to compose directly. This capability and its extensions to more than two redescriptions has been previously studied [Kumar et al. 2006].

With the above formalisms, given a CDM query $Q(E_1, E_2, \ldots, E_k)$ on $\mathcal{S}$ and a subgraph $\Gamma(\mathcal{T})$ of $\Gamma(\mathcal{S})$ matching it, $\Phi(Q, \mathcal{T})$ is a set of bicluster predicates



(a) "Gene properties": composition plan one for the matching subgraph.

(b) "Gene properties": composition plan two for the matching subgraph.

(c) "Gene properties": composition plan three for the matching subgraph.

(d) "Gene properties": composition plan four for the matching subgraph.

Fig. 10. Composition plans for the CDM query in the "Gene properties" example.

$\boldsymbol{\beta} = \{\beta_1, \beta_2, \ldots, \beta_m\}$ and a set of redescription predicates $\boldsymbol{\rho} = \{\rho_1, \rho_2, \ldots, \rho_n\}$ such that

(i) there is a one-to-one correspondence between the bicluster predicates in $\boldsymbol{\beta}$ and the nodes in $\Gamma(\mathcal{T})$.

(ii) for every redescription in $\boldsymbol{\rho}$ there is exactly one edge corresponding to it in $\Gamma(\mathcal{T})$.

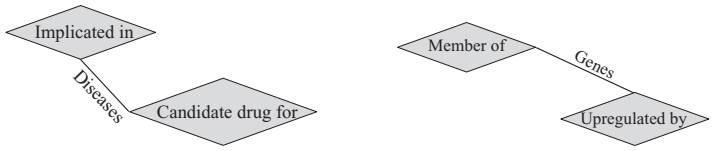
(iii) If a bicluster predicate β_i corresponds to a node in $\Gamma(\mathcal{T})$ and a redescription predicate ρ_j corresponds to an edge incident on that node, then β_i and ρ_j compose.

(iv) the subgraph of $\Gamma(\mathcal{T})$ induced by nodes corresponding to bicluster predicates in $\boldsymbol{\beta}$ and edges corresponding to redescription predicates in $\boldsymbol{\rho}$ is connected.

Note that an edge in this subgraph of $\Gamma(\mathcal{T})$ and the two nodes incident on it correspond to a $\beta\rho\beta$ pattern, reinforcing our decision to use these patterns as the building blocks of CDM. Just as there can be multiple subgraphs matching a CDM query, there can be multiple composition plans corresponding to a $(\boldsymbol{Q}, \Gamma(\mathcal{T}))$ pair. We can graphically depict any plan by highlighting the subgraph of $\Gamma(\mathcal{T})$ corresponding to plan (defined in condition (iv) above). For instance, Figure 10 displays four composition plans for the single subgraph that matches the CDM query for the "Gene properties" example, and Figure 11 displays one composition plan each for the two subgraphs that match the CDM query for the "Small molecules" example.

4. ALGORITHMS FOR CDM

To answer a CDM query, there are three key problems to be solved:



(a) "Small molecules": composition plan for the first subgraph matching the CDM query.

(b) "Small molecules": composition plan for the second subgraph matching the CDM query.

Fig. 11. Composition plans for the CDM query in the "Small molecules" example.

(1) Identify all possible subgraphs of the given database schema that match the query.
(2) For each subgraph, derive all specific composition plans.
(3) For each composition plan, compute all relevant $\beta\rho\beta$ patterns.

We present efficient algorithms for each of these stages. For ease of understanding we present them in the reverse order, so that each algorithm feeds into the input of the next. Note that given an instance of a CDM schema and a composition plan $\Phi(\boldsymbol{Q}, \mathcal{T})$, finding satisfying assignments for $\boldsymbol{\beta}$ and $\boldsymbol{\rho}$ in $\Phi(\boldsymbol{Q}, \mathcal{T})$ reduces to an database join over $\beta\rho\beta$ predicates.

4.1 Computing $\beta\rho\beta$ Patterns

At this stage, we are given two relationships $R_1(D, E)$ and $R_2(E, F)$ that share a common entity set E and a support threshold $k > 0$. Our goal is to compute satisfying assignments for the $\beta_1\rho\beta_2$ pattern, where β_1 (respectively, β_2) is the bicluster predicate corresponding to R_1 (respectively, R_2) and ρ is a redescription predicate between descriptors over E such that the two descriptors participating in ρ contain at least k elements in common.

4.1.1 *Compute then Compose.* In this section, we present a simple algorithm to compute the desired $\beta\rho\beta$ patterns. This approach works by computing all biclusters in R_1 and in R_2 and computing redescriptions between all pairs of projections of these biclusters onto E.

(1) Compute the set of all biclusters in R_1 and in R_2 and their projections onto E.
(2) Insert these projections into a suitable index. Query the index with each projection to compute all its redescriptions.
(3) For each redescription $\rho(X, Y)$ computed in the previous step, let B_1 (respectively, B_2) be the bicluster whose projection onto E is X (respectively, Y). Store the $\beta\rho\beta$ pattern corresponding to this triple.
(4) Return all computed $\beta\rho\beta$ patterns.

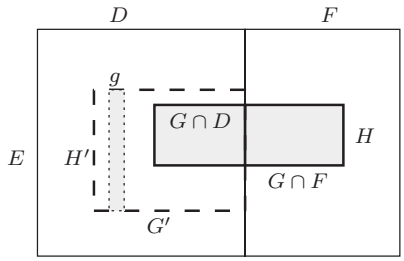


Fig. 12. An illustration of straddling biclusters. The two rectangles with thin borders represent the relationships $R_1(D, E)$ and $R_2(F, E)$. The shaded rectangle with a solid thick border is the straddling bicluster (G, H). The rectangle with a dashed thick border is a closure (G', H') of $(G \cap D, H)$. The dotted rectangle represents the element $g \in D$.

For the purpose of this section, it is enough to assume that the indexing structure simply stores all projections. When given a query projection P, it exhaustively computes all stored projections that contain at least k elements in common with P.

4.1.2 *Compose then Compute.* A concern with the approach just described is that many computed biclusters will not participate in any redescription. In this section, we describe a technique that dramatically reduces the number of such orphan biclusters by mutually biclustering R_1 and R_2.

Let D, E, and F be three entity sets in $\mathcal{E}$ and let $R_1(D, E)$ and $R_2(F, E)$ be two relationships, both connected to the entity set E. Consider the relationship $R_3(D \cup F, E) = R_1(D, E) \cup R_2^T(F, E)$ formed by taking the union of the pairs in the relationships $R_1(D, E)$ and $R_2^T(F, E)$, where the pair (x, y) is a member of $R_2^T(F, E)$ if and only if (y, x) is a member of $R_2(E, F)$. We say that a bicluster (G, H) on $R_3(D \cup F, E)$ *straddles* D and F if G contains at least one element from D and at least one element from F. We define the *component* $B_A(G, H)$ of $B(G, H)$ in A to be the bicluster induced by $G \cap A$ and H on $R(A, B)$. We define the component $B_C(G, H)$ similarly on $R(C, B)$. Note that the components themselves may not be closed. Figure 12 illustrates this situation.

LEMMA 4.1. *Let (G, H) be a closed bicluster on $R_3(D \cup F, E)$ that straddles D and F. Then the closure of the bicluster $(G \cap D, H)$ on R_1 is unique.*

PROOF. Let (G', H') be a closure of $(G \cap D, H)$. By definition of the closure, we have that $G' \supseteq G \cap D$ and $H' \supseteq H$. We will first prove that $G' = G \cap D$. We will then use this constraint to construct a unique H'. Assume to the contrary that there exists an element $g \in D$ that belongs to $G' - G \cap D$. Since (G', H') is a bicluster, for every $h \in H'$, the pair (g, h) is a member of the relationship $R_1(D, E)$. Since $H' \supseteq H$, we see that $(G \cup \{g\}, H)$ is a bicluster on $R_3(D \cup F, E)$, which contradicts the fact that the original bicluster (G, H) is closed. Therefore, $G' = G \cap D$. Now consider an element $e \in E$ such that for all $g \in G \cap D$, the pair (g, e) is a member of the relationship $R_1(D, E)$. By the definition of the closure, H' is the set of all such elements e; H' contains H and is unique. □

This lemma suggests that instead of computing biclusters separately in R_1 and R_2 and subsequently searching for redescriptions between their projections onto E, we can directly compute biclusters with at least k in R_3 and use the closures of its "components" in R_1 and R_2 as seeds for redescription computations. Our modified algorithm to compute $\beta\rho\beta$ patterns has the following steps:

(1) (a) Construct the relationship $R_3(D \cup F, E)$.
    (b) Compute all straddling biclusters in R_3 with at least k elements from E.
    (c) For every bicluster (G, H) computed in Step 1b, compute the closures of the bicluster $(G \cap D, H)$ on R_1 and of the bicluster $(G \cap F, H)$ on R_2.
    (d) Let $\mathcal{P}_1$ (respectively, $\mathcal{P}_2$) denote the set of projections onto E of the closures computed in Step 1c in relationship R_1 (respectively, R_2). Compute all closed biclusters in R_1 (respectively, R_2) with the property that the projection onto E of each such bicluster contains at least one of the projections in $\mathcal{P}_1$ (respectively, $\mathcal{P}_2$).
(2) Identical to Step 2 of the compute then compose algorithm, but applied only to the biclusters computed in Step 1d.
(3)–(4) Identical to Steps 3 and 4 of the compute then compose algorithm.

We now prove that the modified algorithm computes every redescription that the first algorithm does.

LEMMA 4.2. *Let (W, X) be a closed bicluster on R_1 and (Y, Z) be a closed bicluster on R_2^T such that $W \cap Y$ contains at least k elements. Then the algorithm presented above computes the redescription $\rho(X, Y)$.*

PROOF. It is enough to show that the algorithm will compute the two biclusters either in Step 1c or in Step 1d. We will prove that the algorithm will compute (W, X). The proof for (Y, Z) is analogous. Let $U = X \cap Z$.

Assume that there exists a closed bicluster (S, T) on R_3 such that $U \subseteq T \subseteq X$. Since T has at least k elements, the algorithm computes (S, T) in Step 1b. By Lemma 4.1, the closure of $(S \cap D, T)$ is unique. Let this closure be $(S \cap D, T')$. We claim that $T' \subseteq X$. Observe that $S \cap D$ must contain W. Therefore, if T' contains an element $e \notin X$, since e shares a relation with every element of $S \cap D$, e must share a relationship with every element of W, contradicting the fact that (W, X) is closed. Since the algorithm computes (S, T) in Step 1b, it must compute $(S \cap D, T')$ in Step 1c. In other words T' is an element of the set of projections $\mathcal{P}_1$. Since $T' \subseteq X$, we now see the algorithm computes (W, X) in Step 1d.

It remains to show that there exists a closed bicluster (S, T) on R_3 such that $U \subseteq T \subseteq X$. Consider the (possibly nonclosed) bicluster (W, U) on R_1. Consider the closure (W', U') of (W, U) such that $|U' - U|$ is the smallest over all such closures. Clearly, $U' \subseteq X$. Similarly, consider the bicluster (Y, U) on R_2 and its closure (Y', U'') on R_2 such that $|U'' - U|$ is the smallest over all such closures.

Now, $U'' \subseteq Z$. Setting $S = W' \cup Y'$ and $T = U' \cap U''$ yields us the required bicluster. □

As we will show in Section 5, the improved algorithm significantly reduces the number of orphan biclusters while ensuring that we compute exactly the same number of redescriptions.

A final observation is that even for the two given relationships $R_1(D, E)$ and $R_2(E, F)$, there may be multiple $\beta\rho\beta$ patterns possible. If D and E are identical and R_1 is not symmetric, then there are two $\beta\rho\beta$ patterns possible, depending on which "side" of R_1 is used in the redescription with R_2. An example is when R_1 represents genetic interactions where the knock-out of one gene results in a phenotype that enhances or suppresses the phenotype obtained by knocking out the other gene. For such relationships, we define two β predicates for each bicluster, one being the transpose of the other. (Observe that, in addition, if E and F are identical and R_2 is asymmetric, there are four possible $\beta\rho\beta$ patterns.)

4.2 Levelwise Search for Compositional Patterns

We view the "compose then compute" algorithm as an approach to find satisfying assignments for $\beta\rho\beta$ predicates. Then the search for a compositional pattern reduces to relational data mining over the $\beta\rho\beta$ relation. In the following, we will assume that at least two relationships are involved in a compositional pattern (mining one relationship is the task of traditional bicluster mining so that an expressive primitive such as $\beta\rho\beta$ is not required).

In traditional relational mining algorithms such as WARMR [Dehaspe and Toivonen 1999], which support general Datalog queries, the search space of possible patterns is huge, so declarative language biases are imposed. Proteus, too, requires biases to curtail the complexity of search. Before we describe these, it is instructive to examine the structure of a sample composition plan.

Consider the three $\beta\rho\beta$ predicates—$\beta_1\rho_1\beta_2$, $\beta_2\rho_1\beta_3$, and $\beta_1\rho_1\beta_3$—derived from four entity sets, three of whom have binary relationships to the fourth (which supplies the redescription interface ρ_1). Given a CDM query that requires participation of all four entity sets, there are four composition plans possible (the "," denotes conjunction):

—$\beta_1\rho_1\beta_2(X, Y, Z, W)$, $\beta_1\rho_1\beta_3(X, Y, L, M)$.
—$\beta_1\rho_1\beta_2(X, Y, Z, W)$, $\beta_2\rho_1\beta_3(W, Z, L, M)$.
—$\beta_1\rho_1\beta_3(X, Y, L, M)$, $\beta_2\rho_1\beta_3(W, Z, L, M)$.
—$\beta_1\rho_1\beta_2(X, Y, Z, W)$, $\beta_2\rho_1\beta_3(W, Z, L, M)$, $\beta_1\rho_1\beta_3(X, Y, L, M)$.

(We use capital letters denote arguments; recall that they denote sets of objects from the respective domains). Observe the implicit reuse of arguments across predicates, so that the following composition is not legal:

—$\beta_1\rho_1\beta_2(X, Y, Z, W)$, $\beta_1\rho_1\beta_3(R, S, L, M)$.

The typical way in which illegal compositions are avoided is to adopt a canonical ordering for predicates in conjunctive plans and to use *mode* declarations that impose restrictions on how variables are introduced by the predicates. Thus, a

mode of "−" means that the variable can be bound by the predicate itself, "+" means that it must be bound before the predicate is invoked, and "±" means that it can either be bound before or by the predicate. To prevent the above illegal composition, we can specify the mode declarations for the $\beta_1\rho_1\beta_2$ and $\beta_1\rho_1\beta_3$ predicates as

$$—\beta_1\rho_1\beta_2(-, -, -, -)$$
$$—\beta_1\rho_1\beta_3(+, +, -, -)$$

which ensures that the first two arguments of $\beta_1\rho_1\beta_3$ are bound earlier (in this case, by $\beta_1\rho_1\beta_2$). Rather than specify one global set of mode declarations for all compositional patterns, we exploit the fact that the bicluster predicates β_i in the $\beta\rho\beta$s are typed and that every $\beta\rho\beta$ predicate can be used at most once in a composition plan (recall the definition in Section 3.5). With these constraints, it is easy to see that the modes should be "−" for all arguments of the first predicate, and for every predicate following it, use "+" for the mode if the bicluster corresponding to those arguments already participates in a previous $\beta\rho\beta$, and "−" otherwise.

Typical levelwise algorithms used in data mining use the notion of support to prune searches. However, defining a notion of support for CDM patterns is problematic. Due to the multiple shifts of vocabulary that happen in biclusters in a composition, there may be no single domain over which we can define support. It may be possible to define support in database schemas where there is a single domain participating in every relationship. In such a case, since every CDM pattern will involve that domain, we can measure support as the number of entities from that domain that participate in every bicluster in the composition.

A more general approach, used in algorithms such as WARMR [Dehaspe and Toivonen 1999], is to designate a subset of variables as the *key*. The frequency of a pattern is then defined as the number of satisfying assignments to the key for which the pattern is true. This is a natural notion in WARMR whose predicate arguments are individual-based whereas the predicate arguments in Proteus are set-based. A literal mapping of this definition to our relational setting would apply, for instance, if we are seeking "biclusters that participate in at least k compositions." However, the more natural interpretation for biologists is to find "compositions of biclusters and redescriptions that involve at least k (key) objects." (In our applications, the key is typically a central biological object of interest such as genes, or proteins.) In other words, although we have elevated the representation language from objects to sets, data mining constraints are more naturally specified at the object level. Hence, this is the definition we adopt which also affords a levelwise algorithm. In particular, to find compositions of length m that involve at least k objects, we search bottom-up, from level 1 to level $m-1$ for $\beta\rho\beta$s and $\beta\rho\beta$ compositions that involve at least k objects. Due to the anti-monotonicity principle, if a subcomposition does not have support, we need not explore the lattice of $\beta\rho\beta$ patterns that are a superset of the subcomposition. Observe that this allows to 'push' the support constraint into the algorithm for computing $\beta\rho\beta$s, as discussed in the previous section.

Two other considerations are those of logical redundancy of $\beta\rho\beta$ compositions and the specialization relation used to traverse the $\beta\rho\beta$ lattice. Since our compositions are nonrecursive, no redundant compositions should be introduced as long as we adopt a canonical ordering of $\beta\rho\beta$ predicates, such as Rymon's enumeration strategy [Rymon 1992]. However, a more subtle notion of redundancy arises if the original relationship run from an entity set to itself. Consider for instance β_1 derived from a genes-to-genes relationship based on whether their protein products interact, and β_2 derived from a genes-to-genes relationship based on whether the protein product of one transcriptionally regulates the other. In this case, there are two ways in which the biclusters can be related by a redescription, depending on whether the protein interaction relationship extends the transcription regulators or the regulated genes. As mentioned in the previous section, this redundancy is handled at the level of computing $\beta\rho\beta$s itself, so that the notion of strong typing continues to hold when we compose the $\beta\rho\beta$s. The specialization relation is necessary in order to generate candidates. For instance, $\beta_1\rho_1\beta_2(X, Y, Z, W)$ can be specialized to either $\beta_1\rho_1\beta_2(X, Y, Z, W)$, $\beta_1\rho_1\beta_3(X, Y, L, M)$ or to $\beta_1\rho_1\beta_2(X, X, Z, W)$ (the latter makes sense only for symmetric relationships). Again, since $\beta\rho\beta$s are computed by the compose then compute algorithm, we do not have to explicitly search for such assignments. These considerations lead to a straightforward implementation of a levelwise miner along the lines of *Apriori* [Agrawal and Srikant 1994] and WARMR [Dehaspe and Toivonen 1999], which we do not describe in detail in this paper.

4.3 Identifying Matching Subgraphs

Finally, given a CDM query, we address the problem of identifying the relationships and intermediate entity sets that must participate in the composition, which in turn influences the choice of $\beta\rho\beta$s that can be used. The necessary condition here is that the subgraph induced over the database schema should be connected. This is necessary for the $\beta\rho\beta$s to be composable. (It is not sufficient, however, without proper mode declarations, as we saw in the previous section.) If we desire to minimize the number of new entity sets and relationships that are introduced, one possible formulation of this problem is as a computation of a Steiner tree over the database schema. However, cyclicity is not an undesirable feature in a CDM composition and we sometimes might prefer longer compositions, for ease of interpretation. In our current implementation, we exhaustively enumerate all possible subgraphs of the database schema, subject them to membership checks for the domains constrained by the CDM query and, from those that satisfy, identify all the $\beta\rho\beta$s that constitute the subgraph.

5. EFFECTIVENESS OF CDM

Standalone algorithms for redescription mining and biclustering are already heavily tuned. Therefore, the effectiveness of CDM lies in its ability to avoid wasteful computations of biclusters and redescriptions that will not participate in any composition and, for the $\beta\rho\beta$ patterns that remain, being able to

efficiently compose them in the levelwise miner. We have already shown how $\beta\rho\beta$ patterns serve as an important primitive for composition. Hence, in this section we address two questions of algorithmic effectiveness:

(i) What are the savings to computing $\beta\rho\beta$ patterns over separate biclustering and redescription invocations?
(ii) How does the levelwise search for compositions scale with the length of the composition?

We address the first question by assessing, for various pairs of relationships that share a common domain, the number of biclusters that are "orphaned" on either side as a function of the support constraint of the $\beta\rho\beta$ pattern. Figure 13 depicts these plots for various $\beta\rho\beta$ predicates, using relations from a database schema that is described later in Section 6.2. (The exact details of these relations are not as important as the overall trends.) Each plot depicts four curves, two for each bicluster predicate; one curve tracks the number of nonorphan biclusters and the other the number of orphan biclusters, both as functions of the support threshold. Observe that, in general, differences between the number of orphans and the nonorphans can be as great as one to three orders of magnitude. For the plots on the left of Figure 13, for low support thresholds, the number of orphans is smaller than the number of computed biclusters but as the support threshold is increased (number of genes in common, in this case), we see greater numbers of biclusters getting orphaned. For the plots on the right of Figure 13, the number of orphans far exceeds the number of nonorphans, even for low support thresholds. These plots confirm that wasted computation of orphan biclusters is indeed a critical issue in CDM, and highlight the important role played by the compose then compute algorithm developed here.

We study the second question as a function of length of composition, that is, the number of relationships participating in it. Thus, the simplest composition, involving two $\beta\rho\beta$ predicates, has length 3. Again, we use the case study described in Section 6.2 but this time consider the set of all $\beta\rho\beta$ patterns as a whole. We mine $\beta\rho\beta$ patterns at a lenient support constraint of 1. However, even though there is one entity set participating in almost all relationships, we do not impose any support constraints in the levelwise miner. As a result, we may obtain compositions where one set of entities can gradually "morph" into another set of entities without any overlap. Thus, not imposing support constraints allows us to push the levelwise miner to its limits since it may be forced to evaluate a very large number of candidate compositions. Figure 14 (left) displays the number of patterns mined as a function of composition length. Observe that there is initially an increase in number of patterns with length of composition but this number drops off steeply for higher values (there are no patterns mined of composition length 7 or more). It is significant that, for a schema with 9 relationships, we find compositions of length 6 (although not quite evident in Figure 14 (left), there are 45 of them). This statistic demonstrates that there are significant opportunities for CDM in real multirelational datasets. The output-sensitive nature of the levelwise algorithm is evident in Figure 14 (right) which tracks the time taken to mine compositions as a function

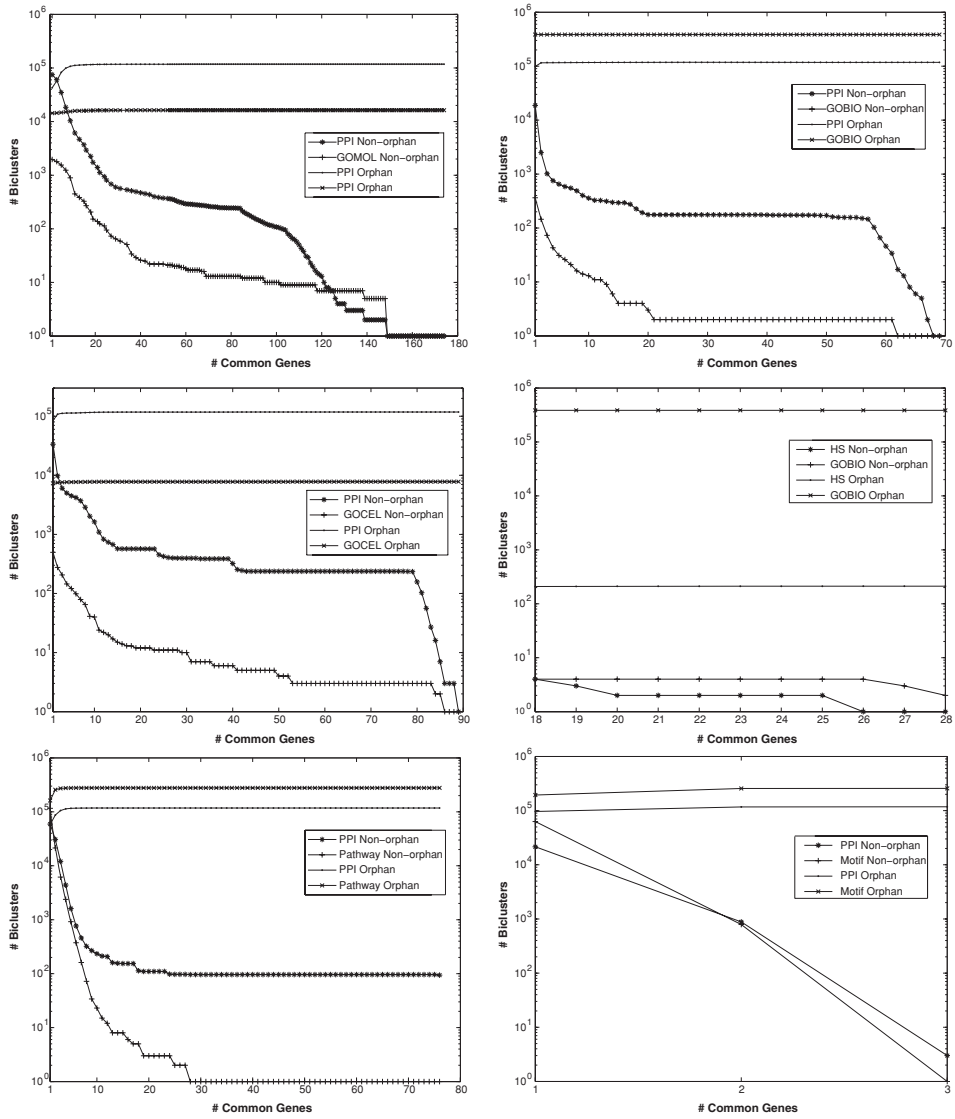


Fig. 13. Assessing the number of "orphan" biclusters avoided as well as the actual biclusters computed (nonorphans) by the "compose then compute" algorithm. Each of the six plots involves a different $\beta\rho\beta$ predicate.

of composition length. (Recall that due to the lax support constraint, the algorithm would be evaluating an exorbitant number of candidates.)

6. CASE STUDIES

Our first case study (GO^3) mines overlaps in functional annotations across all three categories of the Gene Ontology (GO) using human (*H. sapiens*) genes as the underlying universal set. The results of this study help understand implicit

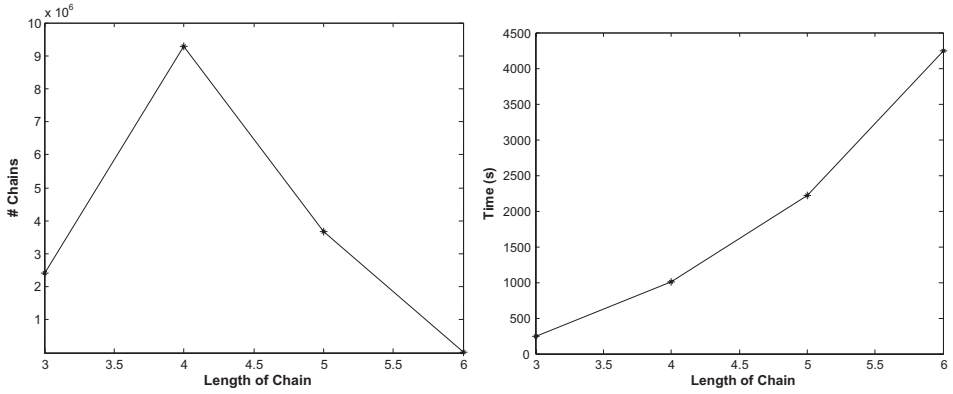


Fig. 14. (left) Number of compositions mined as a function of the length of the composition. (right) Time taken to mine all compositions.

dependencies between terms from different GO categories and potentially to use these dependencies to predict new gene-term associations (an aspect beyond the scope of the present paper). The second case study (Stress Response in Human Cells) focuses on understanding the molecular mechanisms of responses of human cells when they are subjected to different types of environmental stresses. Besides human genes and their membership in GO taxonomies, for this study, we also incorporate data about gene expression measured by microarrays, transcriptional motifs in upstream regions of genes, locations of genes in cytogenetic bands, protein-protein interactions, and pathway membership. Figures 15 and 20 display the schemas for these case studies. In both figures, dashed lines connect pairs of relationships between whose biclusters we compute redescriptions. Table I gives important statistics for both case studies. We provide one table since the data for the second case study subsumes the first.

6.1 GO[3]

The Gene Ontology [Ashburner et al. 2000] is a controlled vocabulary to describe genes and their products across a range of organisms. The three categories of GO—biological process, molecular function, and cellular component—address diverse aspects of gene activity. Briefly, they address the "when," "'what," and "where" of a gene's activity in cells. Each category is organized as a directed acyclic graph (DAG) defined by parent-child relations between terms.

The dependencies we seek to mine are pairs of GO terms, each belonging to a different category, that are annotated by a surprisingly large number of common genes. In this study, each GO term yields exactly one bicluster consisting of that GO term and all the genes annotated with it. Some dependencies are obvious. For instance, we anticipate that the GO biological process "protein ubiquination", the GO molecular function "ubiquitin ligase activity," and the GO cellular component "ubiquitin ligase complex" should annotate nearly the same set of genes. Other such associations might be less obvious, however, and our goal is to mine them.

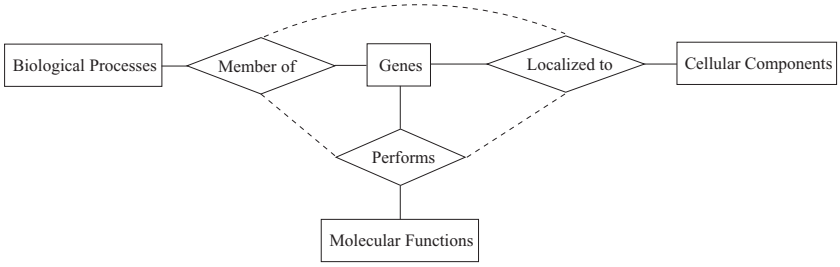


Fig. 15. The schema for the first case study involving GO functional annotations for human genes.

Table I. Statistics for the Two Case Studies.
We only display statistics for relationships involving genes. The first column states the name of the relationship. The second column lists the number of distinct genes participating in the relationship. The third column lists the number of participants from that relationship, whose type is given in the fourth column. The fifth and sixth columns state the number of pairs and density of the relationship. The database contains gene expression measurements for 13 different stresses, each comprising multiple time-points.

Name	#Genes	#Participants	Domain type	#Relationships	Density
PPIs	9318	9318	Genes	45277	0.0005
Gene expression	13877	188	Timepoints	2420842	0.9279
Member of	15498	3307	GO Biological processes	301671	0.0059
Localized to	15498	657	GO Cellular components	171226	0.0168
Performs	15498	2618	GO Molecular functions	152246	0.0038
Member of	13197	1686	MSigDB pathways	106367	0.0048
Contains	9859	837	MSigDB Motifs	101523	0.0123
Belong to	29856	383	MSigDB Cytogenetic bands	60013	0.0052

Since terms in GO are specified at multiple levels of detail, it is not sufficient to evaluate dependencies simply based on the number of genes simultaneously annotating two functions. We use the following strategy, modified from Grossmann et al. [2006]. Given a term s, let n_s be the number of genes annotating the term. Given two terms s and t, let $n_{s,t}$ be the number of genes annotating both terms and $n^+_{s,t}$ be the number of genes annotating at least one parent of either s or t. We want to assess the surprise in observing that s and t annotate $n_{s,t}$ genes in common, conditioned on the fact that their parents annotate $n^+_{s,t}$ genes in total. We ask the following question: if we were to pick n_t genes uniformly at random without replacement from a pool of $n^+_{s,t}$ genes, what is the probability that we will select $n_{s,t}$ or more genes from a set of n_s marked genes? We take recourse to the familiar hypergeometric distribution to assess this probability, denoted $p_{s,t}$:

$$p_{s,t} = \frac{\sum_{k=n_{s,t}}^{\min(n^+_{s,t},n_s)} \binom{n_s}{k}\binom{n^+_{s,t}-n_s}{n_t-k}}{\binom{n^+_{s,t}}{n_t}}.$$

Since we test the significance of multiple pairs of functions, we adjust the p-values using the false discovery rate [Benjamini and Hochberg 1995]. Figure 16 depicts the steep drop in the number of redescriptions that meet increasingly stringent thresholds on either the Jaccard's coefficient or the p-value. We plot

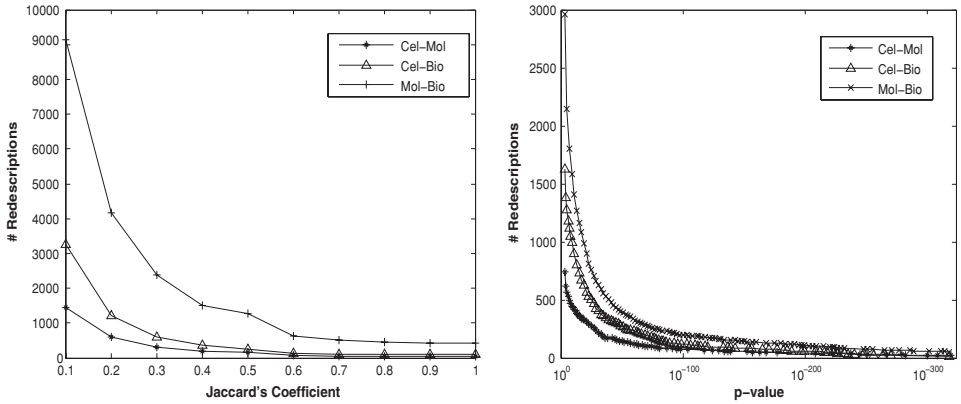


Fig. 16. GO^3 case study: distribution of the number of redescriptions. (left) Number of redescriptions that satisfy different Jaccard's coefficient thresholds. (right) Number of redescriptions that meet different p-value cutoffs.

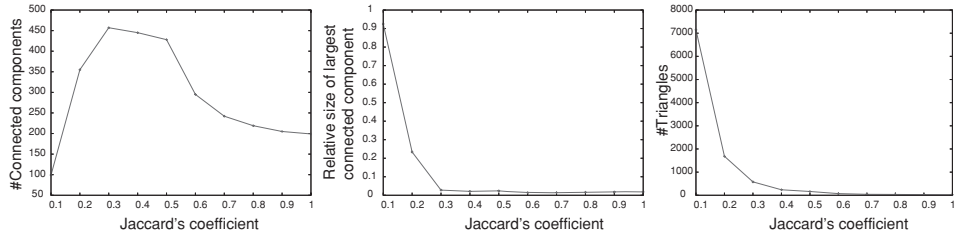
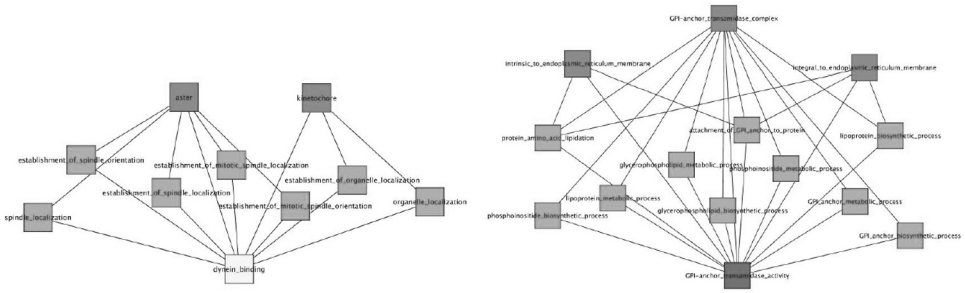


Fig. 17. GO^3 case study: distribution of the number of connected components (left), the relative size of the largest connected component (center), and the number of triangles (right) as a function of Jaccard's coefficient.

separate curves for each pair of GO categories. Observe that the number of redescriptions between GO molecular functions and GO biological processes dominate the number of redescriptions between the other two pairs of categories. This trend reflects the fact that the number of cellular component terms is much smaller than the number of terms in the other two categories (see Table I).

We constructed a graph where each term is a node and two nodes are connected if their redescription is significant at the 0.01 level. By construction, this graph is tripartite. We considered two types of patterns in this graph: triangles and nontriangles. A triangle connects three terms, one from each GO category, such that each pair has significantly overlapping sets of annotated genes. After removing all triangles from this graph, we study the remaining edges that comprise nontriangles. Figure 17 displays global statistics of the structure of this graph as we vary the Jaccard's coefficient. Very few redescriptions satisfy a large Jaccard's coefficient threshold. Therefore, the number of connected components in the graph is small, as is the relative size of the largest component in it and the number of triangles. As we decrease the threshold, more disconnected components start appearing. At a threshold of 0.3, a giant component emerges. As the threshold decreases further, connected components



(a) (b)

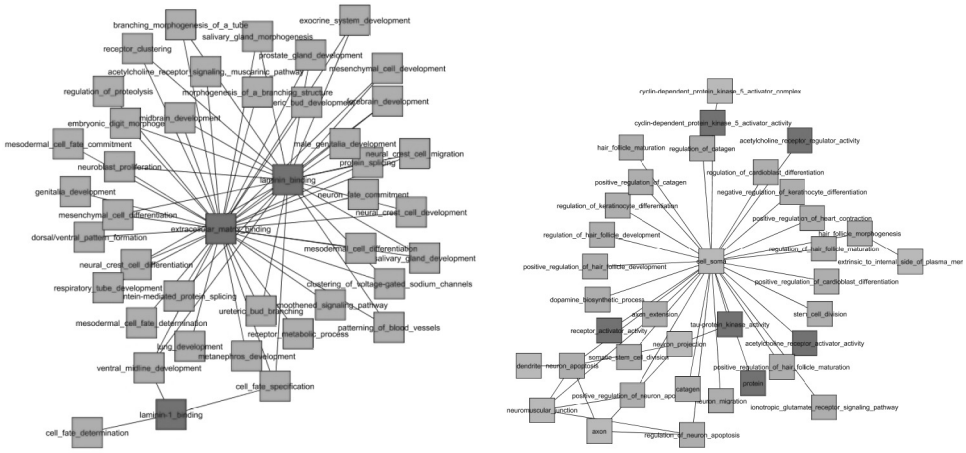
Fig. 18. Examples of triangles in the GO^3 study.

start coalescing. Therefore, the number of connected components decreases. The other two curves are monotonic increasing with decreasing threshold, but show a sharp uptick at 0.3, the point where the giant component forms.

The triangle and nontriangle patterns yielded numerous interesting insights, of which we highlight a few here. In the images we display, each node represents a term in GO (blue nodes are cellular components, green nodes are biological processes, and magenta nodes are molecular functions).

6.1.0.1 *Triangles.* Many triangles represented biological processes fundamental to the function of a cell such as mitosis and important structural components such as the cell membrane. Processes such as mitosis have been studied at depth by biologists. Hence, it is not surprising that the cellular localization of the gene products driving these processes and the molecular functions have been worked out. We hypothesize that a number of annotations for human genes in such triangles are actually electronically transferred from lower organisms such as *S. cerevisiae*. Figure 18(a) displays a subgraph of connected triangles that relate to the process of spindle localization, a key component of cell division. The kinetochore is a protein complex located in the pericentric region of DNA. It provides a point where the microtubules of the spindles can attach. The aster is an array of microtubules that emanate from a spindle pole but do not attach to kinetochores. This subgraph suggests that asters and kinetochores together coordinate the localization of the spindle during cell division. Figure 18(b) displays a network of connected triangles "rooted" at the molecular function "GPI anchor transamidase activity." GPI anchors attach membrane proteins to the cell's lipid bilayer. This subgraph highlights other relevant processes and components involved in this function, for example, the synthesis of phosphoinositides and the GPI anchor transamidase complex.

6.1.0.2 *Nontriangles.* We observed that almost all pairs of terms connected by nontriangle edges related to components, functions, and processes were unique to multicellular and higher order organisms. This observation suggests that such concepts have not been experimentally well-studied in all three categories of GO. Laminins are glycoproteins that are major constituents of the basement membrane of cells. Figure 19(a) demonstrates that the function of



(a) (b)

Fig. 19. Examples of nontriangles in the GO^3 study.

binding with laminins is intimately linked to a very large and diverse set of processes: the development of the prostate and salivary glands, regulation of proteolysis, and cell fate specification (the process involved in the specification of the identity of a cell), to name just a few. Figure 19(b) relates the cell soma, which is the portion of the cell bearing surface projections, to yet another large and diverse set of processes. These processes include stem cell division, regulation of heart contraction, the maturation of hair follicles, and biosynthesis of dopamine.

6.2 Stress Response in Human Cells

Our goal in this case study is to use CDM to understand the cellular contexts in which genes regulated by external stresses operate. We gathered a diverse set of data types to address this question. First, we obtained gene expression data characterizing responses of HeLa cells and primary human lung fibroblasts to heat shock, endoplasmic reticulum stress, oxidative stress, and crowding [Murray et al. 2004]. The dataset we analyzed includes transcriptional measurements obtained by Whitfield et al. [2002] for studying cell cycle arrest by using a double thymidine block or with a thymidine-nocodazole block. Overall, the gene expression data involves 13 distinct stresses over the two cell types. Next, we obtained a network of 31108 molecular interactions between 9243 human gene products by integrating the interactions in the IDSERVE database [Ramani et al. 2005], the results of large scale yeast two-hybrid experiments [Rual et al. 2005; Stelzl et al. 2005], and 20 immune and cancer signalling pathways in the Netpath database (http://www.netpath.org). The IDSERVE database includes human curated interactions from BIND [Bader et al. 2003], HPRD [Peri et al. 2003], and Reactome [Joshi-Tope et al. 2005], interactions predicted based on co-citations in article abstracts, and interactions that transferred from lower eukaryotes based on sequence similarity [Lehner and Fraser

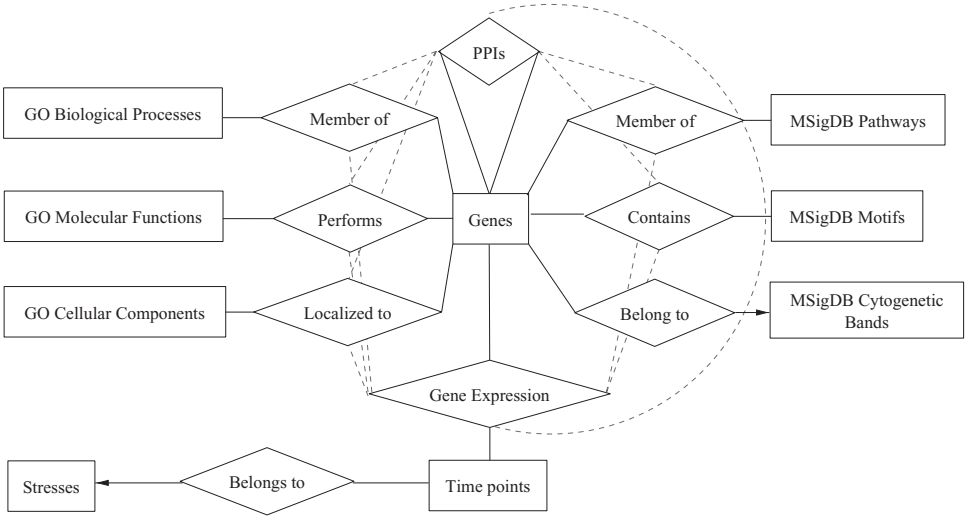


Fig. 20. The schema for the second case study involving human PPIs, stress gene expression data, and MSigDB and GO functional annotations.

2004]. Finally, we derived information about cytogenetic bands, transcriptional motifs, and pathway membership from MSigDB [Subramanian et al. 2005] and functional annotations for the genes in our network from the Gene Ontology (GO) [Ashburner et al. 2000]. Figure 20 displays the database schema underlying this data and Table I summarizes important statistics about this data.

Due to the multitude of data types available, we used a variety of algorithms for computing biclusters. We adapted a home-grown closed itemset mining algorithm to compute straddling biclusters. We used SAMBA [Tanay et al. 2002] to discover biclusters in gene expression data. Since the human PPI network is quite sparse, we found that biclusters in the "PPIs" relationship to be very small in size. Therefore, we simulated the process of redescribing genes in SAMBA biclusters into genes in PPI biclusters by implementing an expansion operator: for each SAMBA bicluster, we constructed a PPI subnetwork that included all genes in that bicluster with known PPIs. We connected pairs of these genes either directly (if they were interacting) or indirectly (if they had a common neighbor). Note that such PPI subnetworks may not be connected. The results we have presented in Section 5 use these biclustering algorithms and expansion operations to showcase the scalability of our CDM implementation for this case study.

A number of compositions we compute illustrate known themes about the cell's response to stress. For instance, it is well known that when targeted by a stress, the cell shuts down the cell cycle in order to cope with the stress. Consistent with this observation, we find that compositions containing SAMBA biclusters with down-regulated genes also involve MSigDB pathways and GO biological processes related to various stages of the cell cycle. In addition SAMBA biclusters with up-regulated genes often compose with MSigDB pathways containing cell cycle regulators.

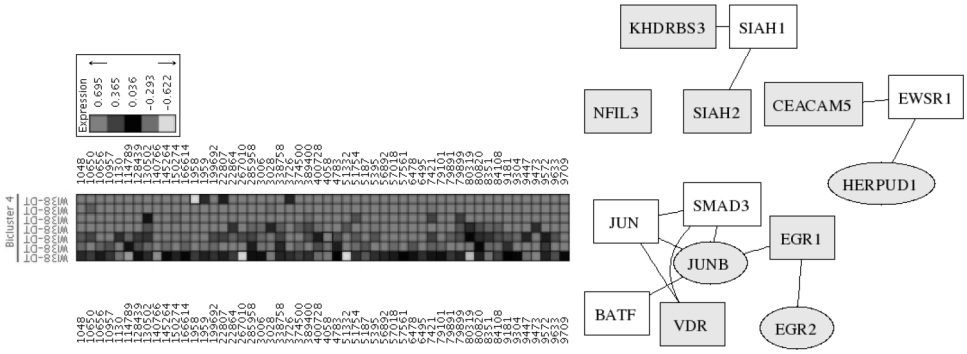


Fig. 21. Stress response in human cells: a CDM pattern that sheds light on fibroblast response to endoplasmic reticulum stress. This pattern involves four relationships: "Gene Expression" (left), "PPIs" (right), "Member of" MSigDB pathways (not shown), and "Belongs to" stresses (not shown). See text for more details.

We highlight a CDM pattern that spans the "Gene Expression," "PPIs," "Member of" (MSigDB pathways), and "Belongs to" (Stresses) relationships, thus connecting four entity sets. The two MSigDB pathways in this pattern are "CMV_HCMV_TIMECOURSE_ALL_UP" and "GALINDO_ACT_UP"; we discuss them in more detail below. This composition involves the response of fibroblasts to treatment with 2.5 mM dithiothreitol (DTT), which is known to induce endoplasmic reticulum stress. The SAMBA bicluster contains six time points (other than the "zero" point), all measuring the response to this stress. All genes in the bicluster are up-regulated, as displayed in Figure 21. The figure also displays the PPI subnetwork corresponding to this bicluster. Here, a light green rectangle is a gene present both in the SAMBA bicluster and the PPI network; a light green ellipse is a gene present in addition in the MSigDB pathways that form this pattern; a white node is one that is introduced by the expansion operator. The "CMV_HCMV_TIMECOURSE_ALL_UP" pathway is a set of 470 genes up-regulated in fibroblasts following infection with human cytomegalovirus [Browne et al. 2001]. The presence of this pathway in this pattern suggests that the endoplasmic reticulum may be targeted by the virus during infection. We find evidence in the literature supporting this CDM pattern. Ogawa-Goto et al. [2002] found that p180, an integral endoplasmic reticulum membrane protein, interacts with a viral protein and that this interaction may play a role in the intracellular transport of the virus. "GALINDO_ACT_UP" is a set of 88 genes significantly up-regulated by the toxin Act in macrophages [Galindo et al. 2003]. This CDM pattern suggests that the inflammatory response induced by this toxin may include stress to the endoplasmic reticulum.

Another pattern spans the same relationships and entity sets. It highlights the response of HeLa cells to oxidative stress induced by administering hydrogen peroxide. As displayed in Figure 22, the genes in the SAMBA bicluster in this composition are heavily down-regulated in response to this treatment. The expanded PPI subnetwork contains a number of proteins involved in apoptosis (programmed cell death). Not surprisingly, one of the MSigDB

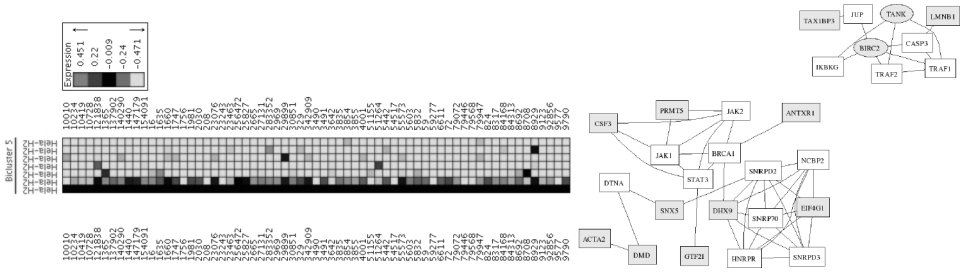


Fig. 22. Stress response in human cells: a CDM pattern that sheds light on Hela cell response to oxidative stress and incipient Alzheimer's disease. This pattern involves four relationships: "Gene Expression" (left), "PPIs" (right), "Member of" MSigDB pathways (not shown), and "Belongs to" stresses (not shown). See text for more details.

pathways participating in this chain is the "CASPASEPATHWAY," which contain proteases active in apoptosis. Another MSigDB pathway that is involved is "HIVNEFPATHWAY," which is the pathway triggered by the HIV-1 protein Nef when it induces the death of T cells. The intriguing aspect of this CDM composition comes from the third MSigDB pathway: "ALZHEIMERS_DISEASE_UP." Microarray analysis defined this set of genes that are up-regulated in incipient Alzheimer's disease [Blalock et al. 2004]. Thus, the activity of these genes in the disease is exactly the opposite of their regulation in response to oxidative stress. This CDM pattern may suggest a potential link between Alzheimer's disease and oxidative stress.

Can CDM patterns be obtained simply by computing functional enrichment? A natural question that arises is whether patterns of the same expressiveness as those in Figures 21 and 22 can be obtained simply by computing the enrichment of the other descriptors in the SAMBA bicluster. We verified that this is not the case in each of the patterns above. Specifically, the p-value of the redescription between the SAMBA bicluster and the MSigDB pathway bicluster is poor (0.01 in the case of the pattern in Figure 21 and 0.9 in the case of the pattern in Figure 22). Therefore, even though the gene interface is shared between the SAMBA and the MSigDB pathway biclusters, we need the intermediate PPI bicluster to form the CDM pattern.

7. RELATED RESEARCH

As proposed here, compositional data mining is a new analysis paradigm that subsumes many data mining formulations such as association rule analysis [Agrawal and Srikant 1994], subspace clustering [Agrawal et al. 2005], inductive logic programming [Dzeroski and Lavrac 2001; Muggleton 1999], and schema matching [Dhamankar et al. 2004; Rahm and Bernstein 2001]. It generalizes association rule mining in that it finds two-way connections between sets of objects, rather than the one-sided implications modeled by associations. It generalizes subspace clustering by identifying concerted subspaces across multiple domains by navigating a general database schema. It generalizes inductive logic programming by finding relational connections not between objects, but between sets of objects. Finally, CDM generalizes schema matching

by uncovering semantic mappings across domains, wherein the 'schemas' are generalized sets, not just attribute-based partitionings.

The compositions computed by Proteus have similarities to the "chains of relations" studied in Afrati et al. [2005]. Here the authors focus on compositions involving two relations and study the problem of finding objects in one relation that, when projected onto the second relation, satisfy a desired property. For properties of the induced graph that satisfy anti-monotonicity constraints, they propose *Apriori*-like algorithms; for other properties, they propose combinatorial optimization algorithms based on integer programming. Our compositions, on the other hand, are based on enumerative generation by following a template rather than finding the "best composition" according to some optimization criteria. It is an aspect of future work to push such constraints into the CDM pipeline, especially to determine suitable abstractions like $\beta\rho\beta$s that can directly yield optimized chains. Furthermore, we consider longer chains and allow greater laxity in how descriptors (called "selectors" in [Afrati et al. 2005]) are defined.

CDM shares many similarities with the "algebra of data mining" recently proposed by Calders et al. [2006]. Their intensional and extensional definitions of "regions" mirror the notion of descriptors, and their "bridges" from a data world to a region world are similar to our mappings between the given database schema and the CDM schema. Using a small set of mining operators, Calders et al. are able to cast many complex data mining scenarios as compositions of their operators. Our work has similar motivations in the compositional approach to data mining and the emphasis on sets of objects. However, the two mining primitives used here are oriented toward supporting arbitrary relational set-based compositions instead of the broad range of mining algorithms studied in Calders et al. [2006]. We also provide efficient algorithmic implementations of CDM whereas the emphasis in Calders et al. [2006] is on studying the complexity of answering different classes of data mining queries.

The use of redescriptions to mediate compositions is similar to "soft joins" as used in the WHIRL system [Cohen 2000] and set-based similarity joins as studied by Sarawagi and Kirpal [2004]. CDM patterns are also similar to the work of Long et al. [2006] who cast it as a problem of finding hidden structures in a multi-partite relation graph. However, the work of Long et al. develops a specialized multiclustering algorithm whereas we compositionally build upon algorithms that work with the individual domains and relationships.

8. DISCUSSION

This article has presented a compositional approach to mining multirelational patterns involving sets and demonstrated its usefulness in two bioinformatics applications. We anticipate that the approach presented here is a start to better conceptualization of biological data mining problems and will spur further development of expressive primitives. Rather than developing special purpose algorithms for every new type of dataset or analysis goal, CDM encourages us to abstract out specifics of different biological contexts and think modularly about analysis objectives. The work proposed here is also a precursor to designing complex data mining applications over large community-maintained

resources, such as SGD [Christie et al. 2004], Wormbase [Chen et al. 2005], FlyBase [Drysdale and Crosby 2005], and TAIR [Huala et al. 2001]. Since many of these resources are typically organized using relational database technology. they constitute a fertile ground for information integration and multirelational knowledge discovery using Proteus.

CDM patterns can be viewed as answered to structured "fill-in-the-blanks" questions. For instance, a biologist desiring to connect genes involved in response to oxidative stress to Alzheimer's disease can use the pattern from Figure 22 to identify the PPIs that might be involved in this connection. Furthermore, among all the possible relationships that could have been used to form this pattern, the fact that the PPI relationship is used suggests that this connection is at the level of protein interactions. Instead, if the intermediate bicluster were formed by MSigDB motifs, that would suggest a relationship at the level of transcriptional regulation. Ongoing work addresses the use of CDM patterns to formulate new hypotheses for specific biological problems.

In future work, we plan to expand the scope of CDM queries to involve arbitrary set constructions as supported by a full-fledged redescription miner such as CARTwheels [Ramakrishnan et al. 2004] or BLOSOM [Zhao et al. 2006]. These, in turn, will require more expressive biclustering algorithms that can accommodate richer constraints and work concertedly with the redescription miner. Finally, although our "compose then compute" approach already avoids wasteful computation of biclusters, for other classes of queries (e.g., one that requests chains involving only a given "seed" set of genes), greater levels of pruning in computed biclusters and redescriptions can be attained. Finally, we aim to support a broader class of queries (e.g., counting and existence checks for compositional patterns) that will support important multirelational knowledge discovery tasks. This will help make compositional data mining as seamless and natural as (compositional) database querying.

REFERENCES

AFRATI, F., DAS, G., GIONIS, A., MANNILA, H., MIELIKAINEN, T., AND TSAPARAS, P. 2005. Mining chains of relations. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM'05)*. 553–556.

AGRAWAL, R., GEHRKE, J., GUNOPULOS, D., AND RAGHAVAN, P. 2005. Automatic subspace clustering of high dimensional data. *Data Min. Knowl. Discov. 11*, 1, 5–33.

AGRAWAL, R. AND SRIKANT, R. 1994. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94)*. 487–499.

ASHBURNER, M., BALL, C. A., BLAKE, J. A., BOTSTEIN, D., BUTLER, H., CHERRY, J. M., DAVIS, A. P., DOLINSKI, K., DWIGHT, S. S., EPPIG, J. T., HARRIS, M. A., HILL, D. P., ISSEL-TARVER, L., KASARSKIS, A., LEWIS, S., MATESE, J. C., RICHARDSON, J. E., RINGWALD, M., RUBIN, G. M., AND SHERLOCK, G. 2000. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nat. Genet. 25*, 1 (May), 25–29.

BADER, G., BETEL, D., AND HOGUE, C. 2003. BIND: the biomolecular interaction network database. *Nucleic Acids Resear. 31*, 1, 248–250.

BALL, C., AWAD, I., DEMETER, J., GOLLUB, J., HEBERT, J., HERNANDEZ-BOUSSARD, T., JIN, H., MATESE, J., NITZBERG, M., WYMORE, F., ZACHARIAH, Z., BROWN, P., AND SHERLOCK, G. 2005. The stanford microarray database accomodates additional microarray platforms and data formats. *Nucleic Acids Resear. 1*, 33(Jan.), D580–D582.

BAYARDO, R. 2002. The many roles of constraints in data mining. *ACM SIGKDD Explorations 4*, 1(June), 1–2.

BENJAMINI, Y. AND HOCHBERG, Y. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Royal Statis. Soc. 57*, 289–300.

BLALOCK, E. M., GEDDES, J. W., CHEN, K. C., PORTER, N. M., MARKESBERY, W. R., AND LANDFIELD, P. W. 2004. Incipient Alzheimer's disease: microarray correlation analyses reveal major transcriptional and tumor suppressor responses. *Proc. Natl. Acad. Sci. 101*, 7, 2173–8.

BROWNE, E. P., WING, B., COLEMAN, D., AND SHENK, T. 2001. Altered cellular mRNA levels in human cytomegalovirus-infected fibroblasts: Viral block to the accumulation of antiviral mRNAs. *J. Virol. 75*, 24, 12319–30.

CALDERS, T., LAKSHMANAN, L., NG, R., AND PAREDAENS, J. 2006. Expressive power of an algebra for data mining. *ACM Trans. Datab. Syst. 31*, 4, 1169–1214.

CARPENTER, A. AND SABATINI, D. 2004. Systematic genome-wide screens of gene function. *Nature Rev. Genetics 5*, 1(Jan.), 11–22.

CHEN ET AL., N. 2005. WormBase: A comprehensive data resource for *Caenorhabditis* biology and genomics. *Nucleic Acids Resear. 33*, D383–D389.

CHRISTIE, K., WENG, S., BALAKRISHNAN, R., COSTANZO, M., DOLINSKI, K., DWIGHT, S., ENGEL, S., FEIERBACH, B., FISK, D., HIRSCHMAN, J., HONG, E., ISSEL-TARVER, L., NASH, R., SETHURAMAN, A., STARR, B., THEESFELD, C., ANDRADA, R., BINKLEY, G., DONG, Q., LANE, C., SCHROEDER, M., BOTSTEIN, D., AND CHERRY, J. 2004. Saccharomyces Genome Database (SGD) provides tools to identify and analyze sequences from Saccharomyces cerevisiae and related sequences s from other organisms. *Nucleic Acids Resear. 32*, D311–4.

COHEN, W. 2000. WHIRL: A word-based information representation language. *Artif. Intell. 118*, 1–2, 163–196.

DEHASPE, L. AND TOIVONEN, H. 1999. Discovery of frequent datalog patterns. *Data Min. Knowl. Discov. 3*, 1, 7–36.

DHAMANKAR, R., LEE, Y., DOAN, A., HALEVY, A., AND DOMINGOS, P. 2004. iMAP: Discovering complex mappings between database schemas. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'04)*. 383–394.

DRYSDALE, R. A. AND CROSBY, M. A. 2005. FlyBase: Genes and gene models. *Nucleic Acids Resear. 33*.

DZEROSKI, S. AND LAVRAC (EDITORS), N. 2001. *Relational Data Mining*. Springer, Berlin, Germany.

FISHER, D. 1987. Knowledge acquisition via incremental conceptual clustering. *Mach. Learn. 2*, 2, 139–172.

GALINDO, C. L., SHA, J., RIBARDO, D. A., FADL, A. A., PILLAI, L., AND CHOPRA, A. K. 2003. Identification of aeromonas hydrophila cytotoxic enterotoxin-induced genes in macrophages using microarrays. *J. Biol. Chem. 278*, 41, 40198–212.

GROSSMANN, S., BAUER, S., ROBINSON, P., AND VINGRON, M. 2006. An improved statistic for detecting over-represented Gene Ontology annotations in gene sets. Lecture Notes in Computer Science, Vol. 3909, 85–98.

GROTHAUS, G., MUFTI, A., AND MURALI, T. 2006. Automatic layout and visualization of biclusters. *Algor. Molec. Biol. Vol. 1*, 15.

GUNSALUS, K. AND PIANO, F. 2005. RNAi as a tool to study cell biology: Building the genome-phenome bridge. *Cur. Opin. Cell Biol. Vol. 17*, 1, 3–8.

HUALA ET AL., E. 2001. The Arabidopsis Information Resource (TAIR): A comprehensive database and Web-based information retrieval, analysis, and visualization system for a model plant. *Nucleic Acids Resear. 29*, 1, 102–105.

JOSHI-TOPE, G., GILLESPIE, M., VASTRIK, I., D'EUSTACHIO, P., SCHMIDT, E., DE BONO, B., JASSAL, B., GOPINATH, G., WU, G., MATTHEWS, L., LEWIS, S., BIRNEY, E., AND STEIN, L. 2005. Reactome: A knowledgebase of biological pathways. *Nucleic Acids Resear. 33*, D428–32.

KUMAR, D., RAMAKRISHNAN, N., HELM, R., AND POTTS, M. 2006. Algorithms for storytelling. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06)*. 604–610.

LAVRAC, N. AND FLACH, P. 2001. An extended transformation approach to inductive logic programming. *ACM Trans. Computat. Logic 2*, 4, 458–494.

LEHNER, B. AND FRASER, A. G. 2004. A first-draft human protein-interaction map. *Genome Biol 5*, 9, R63.

LONG, B., WU, X., ZHANG, Z., AND YU, P. 2006. Unsupervised Learning on K-partite graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06)*. 317–326.

MADEIRA, S. AND OLIVEIRA, A. 2004. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Trans. Computat. Biol. Bioinform. 1*, 1, 24–45.

MATZKE, M. AND BIRCHLER, J. 2005. RNAi-mediated pathways in the nucleus. *Nature Revi. Genetics 6*, 1, 24–35.

MATZKE, M. AND MATZKE, A. 2004. Planting the seeds of a new paradigm. *PLoS Biol. 2*, 5, 0582–0586.

MICHALSKI, R. 1980. Knowledge acquisition through conceptual Clustering: A theoretical framework and algoritha for partitioning data into conjunctive concepts. *Inter. J. Policy Anal. Inform. Syst. 4*, 219–243.

MUGGLETON, S. 1999. Scientific knowledge discovery using inductive logic programming. *Comm. ACM 42*, 11, 42–46.

MURALI, T. AND KASIF, S. 2003. Extracting conserved gene expression motifs from gene expression data. In *Proceedings of the Pacific Symposium on Biocomputing*. 77–88.

MURRAY, J. I., WHITFIELD, M. L., TRINKLEIN, N. D., MYERS, R. M., BROWN, P. O., AND BOTSTEIN, D. 2004. Diverse and specific gene expression responses to stresses in cultured human cells. *Mol Biol Cell 15*, 5, 2361–74.

OGAWA-GOTO, K., IRIE, S., OMORI, A., MIURA, Y., KATANO, H., HASEGAWA, H., KURATA, T., SATA, T., AND ARAO, Y. 2002. An endoplasmic reticulum protein, p180, is highly expressed in human cytomegalovirus-permissive cells and interacts with the tegument protein encoded by UL48. *J Virol 76*, 5, 2350–62.

PARIDA, L. AND RAMAKRISHNAN, N. 2005. Redescription mining: Structure theory and algorithms. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI'05)*. 837–844.

PATI, A., VASQUEZ-ROBINET, C., HEATH, L., GRENE, R., AND MURALI, T. 2006. XcisClique: Analysis of regulatory bicliques. *BMC Bioinform. Vol. 7*, 1, 218.

PERI, S., NAVARRO, J., AMANCHY, R., KRISTIANSEN, T., JONNALAGADDA, C., SURENDRANATH, V., NIRANJAN, V., MUTHUSAMY, B., GANDHI, T., GRONBORG, M., IBARROLA, N., DESHPANDE, N., SHANKER, K., SHIVASHANKAR, H., RASHMI, B., RAMYA, M., ZHAO, Z., CHANDRIKA, K., PADMA, N., HARSHA, H., YATISH, A., KAVITHA, M., MENEZES, M., CHOUDHURY, D., SURESH, S., GHOSH, N., SARAVANA, R., CHANDRAN, S., KRISHNA, S., JOY, M., ANAND, S., MADAVAN, V., JOSEPH, A., WONG, G., SCHIEMANN, W., CONSTANTINESCU, S., HUANG, L., KHOSRAVI-FAR, R., STEEN, H., TEWARI, M., GHAFFARI, S., BLOBE, G., DANG, C., GARCIA, J., PEVSNER, J., JENSEN, O., ROEPSTORFF, P., DESHPANDE, K., CHINNAIYAN, A., HAMOSH, A., CHAKRAVARTI, A., AND PANDEY, A. 2003. Development of human protein reference database as an initial platform for approaching systems biology in humans. *Genome Res. 13*, 10, 2363–71.

RAHM, E. AND BERNSTEIN, P. 2001. A survey of approaches to automatic schema matching. *VLDB J. 10*, 4, 334–350.

RAMAKRISHNAN, N., KUMAR, D., MISHRA, B., POTTS, M., AND HELM, R. 2004. Turning CARTwheels: An alternating algorithm for mining redescriptions. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*. 266–275.

RAMANI, A. K., BUNESCU, R. C., MOONEY, R. J., AND MARCOTTE, E. M. 2005. Consolidating the set of known human protein-protein interactions in preparation for large-scale mapping of the human interactome. *Genome Biol. 6*, 5, R40.

RUAL ET AL., J. 2005. Towards a proteome-scale map of the human protein-protein interaction network. *Nature 437*, 7062, 1173–1178.

RYMON, R. 1992. Search through systematic set enumeration. In *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*. 539–550.

SARAWAGI, S. AND KIRPAL, A. 2004. Efficient set joins on similarity predicates. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'04)*. 743–754.

STELZL, U., WORM, U., LALOWSKI, M., HAENIG, C., BREMBECK, F., GOEHLER, H., STROEDICKE, M., ZENKNER, M., SCHOENHERR, A., KOEPPEN, S., TIMM, J., MINTZLAFF, S., ABRAHAM, C., BOCK, N., KIETZMANN, S., GOEDDE, A., TOKSOZ, E., DROEGE, A., KROBITSCH, S., KORN, B., BIRCHMEIER, W., LEHRACH, H., AND

WANKER, E. 2005. A human protein-protein interaction network: A resource for annotating the proteome. *Cell 122*, 6, 957–968.

SUBRAMANIAN, A., TAMAYO, P., MOOTHA, V., MUKHERJEE, S., EBERT, B., GILLETTE, M., PAULOVICH, A., POMEROY, S., GOLUB, T., LANDER, E., AND MESIROV, J. 2005. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci*.

TANAY, A., SHARAN, R., AND SHAMIR, R. 2002. Discovering statistically significant biclusters in gene expression data. *Bioinform. 18*, S136–S144.

TANAY, A., SHARAN, R., AND SHAMIR, R. 2005. Biclustering algorithms: A survey. In *Handbook of Computational Molecular Biology*, S. Aluru, Ed. CRC Computer and Information Science Series. Chapman & Hall.

TSUR, D., ULLMAN, J., ABITEBOUL, S., CLIFTON, C., MOTWANI, R., NESTOROV, S., AND ROSENTHAL, A. 1998. Query flocks: A generalization of association rule mining. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'98)*. 1–12.

WHITFIELD, M. L., SHERLOCK, G., SALDANHA, A. J., MURRAY, J. I., BALL, C. A., ALEXANDER, K. E., MATESE, J. C., PEROU, C. M., HURT, M. M., BROWN, P. O., AND BOTSTEIN, D. 2002. Identification of genes periodically expressed in the human cell cycle and their expression in tumors. *Mol. Biol. Cell. 13*, 6, 1977–2000.

ZAKI, M. AND HSIAO, C.-J. 2002. CHARM: An efficient algorithm for closed itemset mining. In *SIAM International Conference on Data Mining*. 457–473.

ZAKI, M. AND RAMAKRISHNAN, N. 2005. Reasoning about sets using redescription mining. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'05)*. 364–373.

ZHAO, L., ZAKI, M., AND RAMAKRISHNAN, N. 2006. BLOSOM: A framework for mining arbitrary boolean expressions over attribute sets. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06)*. 827–832.

Received June 2007; revised November 2007; accepted December 2007