Knowledge Discovery and Data Mining:
Challenges and Realities

Xingquan Zhu
Florida Atlantic University, USA

Ian Davidson
University of Albany, State University of New York, USA

INFORMATION SCIENCE REFERENCE
Hershey · New York

Acquisitions Editor: Kristin Klinger
Development Editor: Kristin Roth
Senior Managing Editor: Jennifer Neidig
Managing Editor: Sara Reed
Assistant Managing Editor: Sharon Berger
Copy Editor: April Schmidt and Erin Meyer
Typesetter: Jamie Snavely
Cover Design: Lisa Tosheff
Printed at: Yurchak Printing Inc.

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue, Suite 200
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@idea-group.com
Web site: http://www.info-sci-ref.com

and in the United Kingdom by
Information Science Reference (an imprint of IGI Global)
3 Henrietta Street
Covent Garden
London WC2E 8LU
Tel: 44 20 7240 0856
Fax: 44 20 7379 0609
Web site: http://www.eurospanonline.com

Library of Congress Cataloging-in-Publication Data

Knowledge discovery and data mining : challenges and realities / Xingquan Zhu and Ian Davidson, editors.

p. cm.

Summary: "This book provides a focal point for research and real-world data mining practitioners that advance knowledge discovery from low-quality data; it presents in-depth experiences and methodologies, providing theoretical and empirical guidance to users who have suffered from underlying low-quality data. Contributions also focus on interdisciplinary collaborations among data quality, data processing, data mining, data privacy, and data sharing"--Provided by publisher.

Includes bibliographical references and index.

ISBN 978-1-59904-252-7 (hardcover) -- ISBN 978-1-59904-254-1 (ebook)

1. Data mining. 2. Expert systems (Computer science) I. Zhu, Xingquan, 1973- II. Davidson, Ian, 1971-

QA76.9.D343K55 2007

005.74--dc22

2006033770

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Chapter X
Knowledge Discovery in Biomedical Data Facilitated by Domain Ontologies

Amandeep S. Sidhu
Curtin University of Technology, Australia

Paul J. Kennedy
University of Technology Sydney, Australia

Simeon Simoff
University of of Western Sydney, Australia

Tharam S. Dillon
Curtin University of Technology, Australia

Elizabeth Chang
Curtin University of Technology, Australia

ABSTRACT

In some real-world areas, it is important to enrich the data with external background knowledge so as to provide context and to facilitate pattern recognition. These areas may be described as data rich but knowledge poor. There are two challenges to incorporate this biological knowledge into the data mining cycle: (1) generating the ontologies; and (2) adapting the data mining algorithms to make use of the ontologies. This chapter presents the state-of-the-art in bringing the background ontology knowledge into the pattern recognition task for biomedical data.

INTRODUCTION

Data mining is traditionally conducted in areas where data abounds. In these areas, the task of the data mining is to identify patterns within the data, which may eventually become knowledge. To this end, the data mining methods used, such as cluster analysis, link analysis and classifica-

tion and regression, typically aim to reduce the amount of information (or data) to facilitate this pattern recognition. These methods do not tend to contain (or bring to the problem) specific domain specific information. In this way, they may be termed "knowledge-empty." However, in some real-world areas, it is important to enrich the data with external background knowledge so as to provide context and to facilitate pattern recognition. These areas may be described as data rich but knowledge poor. External background information that may be used to enrich data and to add context information, and facilitate data mining is in the form of ontologies, or structured vocabularies. So long as the original data can be linked to terms in the ontology, the ontology may be used to provide the necessary knowledge to explain the results and even generate new knowledge.

In accelerating quest for disease biomarkers, the use of high-throughput technologies, such as DNA microarrays and proteomics experiments, has produced vast datasets identifying thousands of genes whose expression patterns differ in diseased vs. normal samples. Although many of these differences may reach statistical significance, they are not biologically meaningful. For example, reports of mRNA or protein changes of as little as two-fold are not uncommon, and although some changes of this magnitude turn out to be important, most are attributes to disease-independent differences between the samples. Evidence gleaned from other studies linking genes to disease is helpful, but with such large datasets, a manual literature review is often not practical. The power of these emerging technologies—the ability to quickly generate large sets of data—has challenged current means of evaluating and validating these data. Thus, one important example of a data rich but knowledge poor area is biological sequence mining. In this area, there exist massive quantities of data generated by the data acquisition technologies. The bioinformatics solutions addressing these data are a major current challenge. However, domain specific ontologies such as gene ontology (GO Consortium, 2001), MeSH (Nelson & Schopen, 2004) and protein ontology (Sidhu & Dillon, 2005a, 2006a) exist to provide context to this complex real world data.

There are two challenges to incorporate this biological knowledge into the data mining cycle: (1) generating the ontologies; and (2) adapting the data mining algorithms to make use of the ontologies. This chapter presents the state-of-the-art in bringing the background ontology knowledge into the pattern recognition task for biomedical data. These methods are also applicable to other areas where domain ontologies are available, such as text mining and multimedia and complex data mining.

GENERATING OONTOLOGIES: CASE OF PROTEIN ONTOLOGY

This section is devoted to the practical aspects of generating ontologies. It presents the work on building the protein ontology (Sidhu et al., 2006a; Sidhu & Dillon, 2005a, 2006b; Sidhu et al., 2005b) in the section "Protein Ontology (PO)." It then compares the structures of the protein ontology and the well established gene ontology (GO Consortium, 2001) in the section "Comparing PO and GO."

Protein Ontology (PO)

Advances in technology and the growth of life sciences are generating ever increasing amounts of data. High-throughput techniques are regularly used to capture thousands of data points in an experiment. The results of these experiments normally end up in scientific databases and publications. Although there have been concerted efforts to capture more scientific data in specialist databases, it is generally acknowledged that only 20% of biological knowledge and data is available in a structured format. The remaining 80% of biological information is hidden in the unstruc-

tured scientific results and texts. Protein ontology (PO) (Sidhu et al., 2006a; Sidhu et al., 2006b; Sidhu et al., 2005a; Sidhu et al., 2005b) provides a common structured vocabulary for this structured and unstructured information and provides researchers a medium to share knowledge in proteomics domain. It consists of concepts, which are data descriptors for proteomics data and the relations among these concepts. Protein ontology has (1) a hierarchical classification of concepts represented as classes, from general to specific; (2) a list of attributes related to each concept, for each class; and (3) a set of relations between classes to link concepts in ontology in more complicated ways then implied by the hierarchy, to promote reuse of concepts in the ontology. Protein ontology provides description for protein domains that can be used to describe proteins in any organism. Protein ontology framework describes: (1) protein sequence and structure information, (2) protein folding process, (3) cellular functions of proteins, (4) molecular bindings internal and external to proteins and (5) constraints affecting the final protein conformation. Protein ontology uses all relevant protein data sources of information. The structure of PO provides the concepts necessary to describe individual proteins, but does not contain individual protein themselves. Files using Web ontology language (OWL) format based on PO acts as instance store for the PO. PO uses data sources include new proteome information resources like PDB, SCOP, and RESID as well as classical sources of information where information is maintained in a knowledge base of scientific text files like OMIM and from various published scientific literature in various journals. PO database is represented using OWL. PO database at the moment contains data instances of following protein families: (1) prion proteins, (2) B.Subtilis, (3) CLIC and (4) PTEN. More protein data instances will be added as PO is more developed. The complete class hierarchy of protein ontology (PO) is shown in Figure 1. More details about PO is available at the Web site: http://www.proteinontology.info/

Semantics in protein data is normally not interpreted by annotating systems, since they are not aware of the specific structural, chemical and cellular interactions of protein complexes. Protein ontology framework provides specific set of rules to cover these application specific semantics. The rules use only the relationships whose semantics are predefined to establish correspondence among terms in PO. The set of relationships with predefined semantics is: {SubClassOf, PartOf, AttributeOf, InstanceOf, and ValueOf}. The PO conceptual modelling encourages the use of strictly typed relations with precisely defined

Figure 1. Class hierarchy of protein ontology

- ProteinOntology
  - AtomicBind
  - Atoms
  - Bind
  - Chains
  - Family
  - ProteinComplex
    - ChemicalBonds
      - CISPeptide
      - DisulphideBond
      - HydrogenBond
      - ResidueLink
      - SaltBridge
    - Constraints
      - GeneticDefects
      - Hydrophobicity
      - ModifiedResidue
    - Entry
      - Description
      - Molecule
      - Reference
    - FunctionalDomains
      - ActiveBindingSites
      - BiologicalFunction
        - PathologicalFunctions
        - PhysiologicalFunctions
      - SourceCell
    - StructuralDomains
      - Helices
        - Helix
          - HelixStructure
      - OtherFolds
        - Turn
          - TurnStructure
      - Sheets
        - Sheet
          - Strands
    - Structure
      - ATOMSequence
      - UnitCell
  - Residues
  - SiteGroup

semantics. Some of these relationships (like SubClassOf, InstanceOf) are somewhat similar to those in RDF schema but the set of relationships that have defined semantics in our conceptual PO model is small so as to maintain simplicity of the system. The following is a description of the set of predefined semantic relationships in our common PO conceptual model.

- **SubClassOf:** The relationship is used to indicate that one concept is a subclass of another concept, for instance: SourceCell SubClassOfFunctionalDomains. That is any instance of SouceCell class is also instance of FunctionalDomains class. All attributes of FunctionalDomains class (_FuncDomain_ Family, _FuncDomain_SuperFamily) are also the attributes of SourceCell class. The relationship SubClassOf is transitive.
- **AttrributeOf:** This relationship indicates that a concept is an attribute of another concept, for instance: _FuncDomain_Family AttributeOf Family. This relationship also referred as PropertyOf, has same semantics as in object-relational databases.
- **PartOf:** This relationship indicates that a concept is a part of another concept, for instance: Chain PartOf ATOMSequence indicates that Chain describing various residue sequences in a protein is a part of definition of ATOMSequence for that protein.
- **InstanceOf:** This relationship indicates that an object is an instance of the class, for instance: ATOMSequenceInstance_10 InstanceOf ATOMSequence indicates that ATOMSequenceInstance_10 is an instance of class ATOMSequence.
- **ValueOf:** This relationship is used to indicate the value of an attribute of an object, for instance: "Homo Sapiens" ValueOf OrganismScientific. The second concept, in turn has an edge, OrganismScientific AttributeOf Molecule, from the object it describes.

Comparing PO and GO

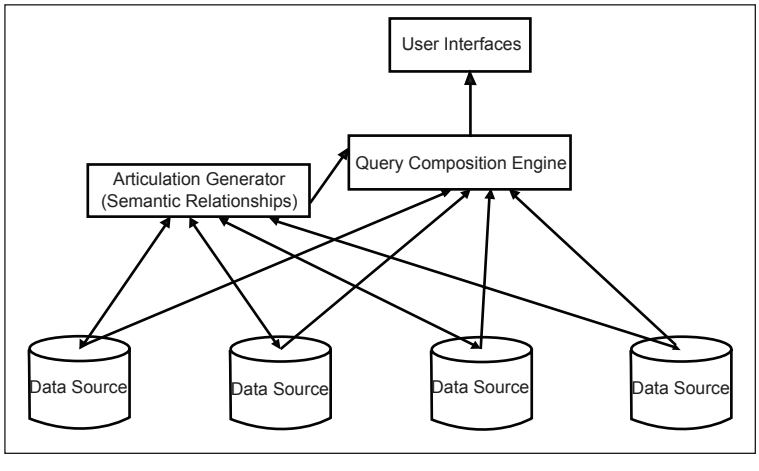
Gene ontology (GO Consortium, 2001) defines a structured controlled vocabulary in the domain of biological functionality. GO initially consisted of a few thousand terms describing the genetic workings of three organisms and was constructed for the express purpose of database interoperability; it has since grown to a terminology of nearly 16,000 terms and is becoming a de facto standard for describing functional aspects of biological entities in all types of organisms. Furthermore, in addition to (and because of) its wide use as a terminological source for database-entry annotation, GO has been used in a wide variety of biomedical research, including analyses of experimental data (GO Consortium, 2001) and predictions of experimental results (GO Consortium & Lewis, 2004)). Characteristics of GO that we believe are most responsible for its success: community involvement; clear goals; limited scope; simple, intuitive structure; continuous evolution; active curation; and early use.

It is clear that organisms across the spectrum of life, to varying degrees, possess large numbers of gene products with similar sequences and roles. Knowledge about a given gene product (i.e., a biologically active molecule that is the deciphered end product of the code stored in a gene) can often be determined experimentally or inferred from its similarity to gene products in other organisms. Research into different biological systems uses different organisms that are chosen because they are amenable to advancing these investigations. For example, the rat is a good model for the study of human heart disease, and the fly is a good model to study cellular differentiation. For each of these model systems, there is a database employing curators who collect and store the body of biological knowledge for that organism. This enormous amount of data can potentially add insight to related molecules found in other organisms. A reliable wet-lab biological experiment performed

in one organism can be used to deduce attributes of an analogous (or related) gene product in another organism, thereby reducing the need to reproduce experiments in each individual organism (which would be expensive, time-consuming, and, in many organisms, technically impossible). Mining of scientific text and literature is done to generate list of keywords that is used as GO terms. However, querying heterogeneous, independent databases in order to draw these inferences is difficult: The different database projects may use different terms to refer to the same concept and the same terms to refer to different concepts. Furthermore, these terms are typically not formally linked with each other in any way. GO seeks to reveal these underlying biological functionalities by providing a structured controlled vocabulary that can be used to describe gene products, and shared between biological databases. This facilitates querying for gene products that share biologically meaningful attributes, whether from separate databases or within the same database.

Challenges faced while developing GO from unstructured and structured data sources are addressed while developing PO. Protein ontology is a conceptual model that aim to support consistent and unambiguous knowledge sharing and that provide a framework for protein data and knowledge integration. PO links concepts to their interpretation, that is, specifications of their meanings including concept definitions and relationships to other concepts. Apart from semantic relationships defined in "Protein Ontology (PO)," PO also model relationships like sequences. By itself semantic relationships described in "Protein Ontology (PO)" does not impose order among the children of the node. In applications using protein sequences, the ability of expressing the order is paramount. Generally protein sequences are a collection of chains of sequence of residues, and that is the format protein sequences have been represented unit now using various data representations and data mining techniques for bioinformatics. When we are defining sequences for semantic heterogeneity of protein data sources using PO we are not only considering traditional representation of protein sequences but also link protein sequences to protein structure, by linking chains of residue sequences to atoms defining three-dimensional structure. In this section we will describe how we used a special semantic relationship like *Sequence(s)* in protein ontology to describe complex concepts defining structure, structural folds and domains and chemical bonds describing protein complexes. PO defines these

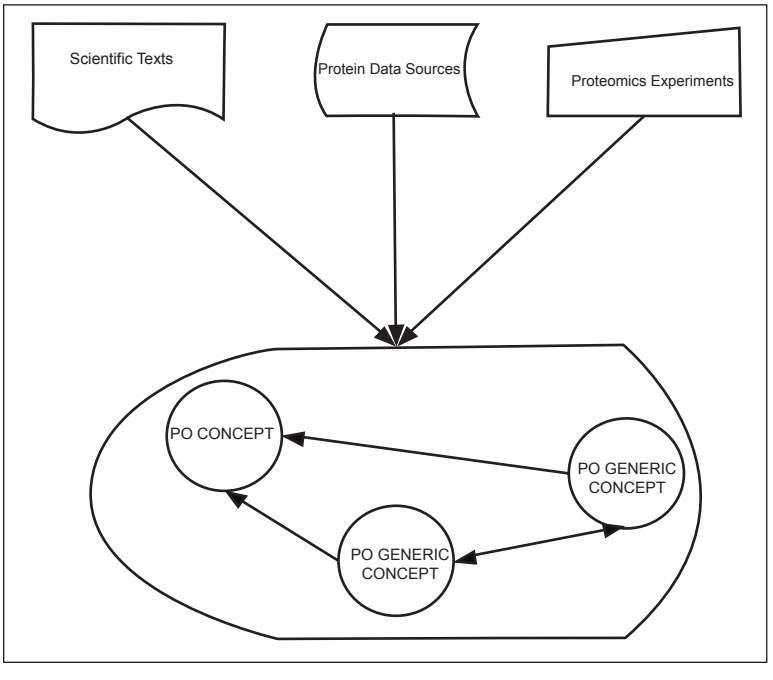
Figure 2. Semantic interoperability framework for PO



complex concepts as *Sequences* of simpler generic concepts defined in PO. These simple concepts are *Sequences* of object and data type properties defining them. A typical example of *Sequence* is as follows. PO defines a complex concept of *ATOMSequence* describing three dimensional structure of protein complex as a combination of simple concepts of *Chains*, *Residues*, and *Atoms* as: *ATOMSequence Sequence (Chains Sequence (Residues Sequence (Atoms)))*. Simple concepts defining ATOMSequence are defined as: *Chains Sequence (ChainID, ChainName, ChainProperty)*; *Residues Sequence (ResidueID, ResidueName, ResidueProperty)*; and *Atoms Sequence (AtomID, Atom, ATOMResSeqNum, X, Y, Z, Occupancy, TempratureFactor, Element)*. Semantic interoperability framework used in PO is depicted Figure 2.

Therefore, PO reflects the structure and relationships of protein data sources. PO removes the constraints of potential interpretations of terms in various data sources and provides a structured vocabulary that unifies and integrates all data and knowledge sources for proteomics domain (Figure 3). There are seven subclasses of protein ontology (PO), called generic classes that are used to define complex concepts in other PO classes: *Residues*, *Chains*, *Atoms*, *Family*, *AtomicBind*, *Bind*, and *SiteGroup*. Concepts from these generic classes are reused in various other PO classes for definition of class specific concepts. Details and properties of residues in a protein sequence are defined by instances of *Residues* class. Instances of chains of residues are defined in *Chains* class. All the three dimensional structure data of protein atoms is represented as instances of *Atoms* class. Defining *Chains*, *Residues* and *Atoms* as individual classes has the benefit that any special properties or changes affecting a particular chain, residue and atom can be easily added. Protein *Family* class represents protein super family and family details of proteins. Data about binding atoms in chemical bonds like hydrogen bond, residue links, and salt bridges is entered into ontology as an instance of *AtomicBind* Class. Similarly the data about binding residues in chemical bonds like disulphide

Figure 3. Unification of protein data and knowledge



bonds and CIS peptides is entered into ontology as an instance of *Bind* class. All data related to site groups of the active binding sites of proteins is defined as instances of *SiteGroup* class. In PO the notions classification, reasoning, and consistency are applied by defining new concepts or classes from defined generic concepts or classes. The concepts derived from generic concepts are placed precisely into class hierarchy of protein ontology to completely represent information defining a protein complex.

As such PO can be used to support automatic semantic interpretation of data and knowledge sources, thus providing a basis for sophisticated mining of information.

CLUSTERING FACILITATED BY DOMAIN ONTOLOGIES

In this section we demonstrate how to modify clustering algorithms in order to utilise the structure of ontology. In the section "Challenges with Clustering Data Enriched with Ontological Information" we present the differences between clustering data items with associated ontological information compared to clustering data items without this information. In "A Distance Function for Clustering Ontologically Enriched Data," we show how these differences must be met in a clustering algorithm. Finally, "Automatic Cluster Identification and Naming" describes an automatic method of naming and describing the clusters found with the domain ontology.

Challenges with Clustering Data Enriched with Ontological Information

Many algorithms exist for clustering data (Duda, Hart, & Stork, 2001; Theodoridis & Koutroumbas, 1999). However, one of the primary decisions to be made when applying cluster analysis to data, and before choosing a specific algorithm, is the way of measuring distances between data items. Generally this involves defining some distance or similarity measure between data items defined in terms of their attributes.

Just as many clustering algorithms have been defined over a wide variety of data types, so to has a large set of potential similarity and distance functions been devised for comparing data items (Theodoridis & Koutroumbas, 1999). In general, a similarity function measures the degree to which two items are similar to one another. Conversely, a distance function measures how two data items are dissimilar. The choice of distance function for data items is often orthogonal to the particular clustering algorithm used as many clustering algorithms take as input a distance matrix, which contains the results of applying a distance function to each combination of data items. The distance matrix is a square symmetric matrix with each cell i, j measuring the distance between data items i and j. The particular distance function used with data items is generally dependent on the type of data being compared. For example, the distance between vectors of real valued data is often defined with the Euclidean distance function, whereas more elaborate functions are required for the sequence data types often found in biomedical datasets.

Thus, the first question we must address when devising a distance function for data enriched with information from ontologies is: what form does the data take? Details will, of course, depend on the particular ontology applied to the data. However, we can make some general comments and apply them in an example of comparing genes based on the associated gene ontology terms. In this example, the "knowledge-poor" or raw data items consist solely of gene names, for example, AA458965 or AA490846, using the GenBank accession codes. These gene names are essentially class labels with no knowledge embedded in them. Hence, there is no useful way to compare them on their own. Ontological information from the

gene ontology may be associated with each gene by using the gene ontology database or with the use of a search engine such as SOURCE (Diehn et al., 2003). In our example, the gene ontology associations are shown in Table 1.

Two characteristics regarding the enriched data are apparent. First, there are different numbers of terms from the gene ontology associated with each gene. For the first gene there are four terms whilst the second has six associations. In general, this will be the norm for associations. Second, we do not seem to have accomplished much from the data enrichment. The associated terms can still be regarded as individual class labels for a very large number of classes (more than 16,000). The terms only have meaning in their relationships within the ontology hierarchy.

Thus, algorithms to cluster ontology enriched data items (1) must be able to handle different numbers of terms associated with data items; and (2) must be able to compare terms based on relationships in the ontology.

A Distance Function for Clustering Ontologically Enriched Data

Given the requirements for the clustering of ontologically enriched data developed in the last section, what kind of similarity measure or distance measure is appropriate? Standard measures like Euclidean distance are not applicable because the data contains different numbers of attributes and there is no natural way to define a distance between classes.

One possible approach is suggested by an analogy to comparison of documents in the field of computational linguistics. A common approach in this field is to transform a free form document into a sparse vector of word counts where each position in the vector refers to a different word in the corpus (see, e.g., Chapter 10 of Shawe-Taylor & Cristianini, 2004). This simplified knowledge representation of the text document ignores relationships between words. In the same way that this representation views a document as a vector of word counts, the ontologically enriched data items may be thought of as a vector of occurrences of gene ontology terms. We could devise a long sparse binary vector with each position referring to the presence or absence of an association with each of the thousands of gene ontology terms to the data item. The problem with this knowledge representation is that most of the gene ontology terms apply to only a very few genes in the database. This means that very few similarities could be found between the vectors for different data items. The solution to this difficulty lies in incorporating the relationships within the ontology into the knowledge representation.

Referring back to the example of the two genes in the last section, there is another characteristic of the enriched data that are not, at first, apparent. We can retrieve further enriched data for the genes by tracing back up the gene ontology hierarchy. In the gene ontology, parent terms are more general concepts of child terms. For example, for the gene AA458965 the term GO:0006952 (defense response) can be derived from the term GO:0006955 (immune response) by following the is-a relationship in the ontology. This allows us to retrieve more general terms describing the genes. These more general terms give a sort of

Table 1. Enriched data. First column lists "knowledge-poor" data in the form of GenBank identifiers. Second column lists associated Gene Ontology term identifiers for each gene.

AA458965	GO:0005125, GO:0005615, GO:0006955, GO:0007155
AA490846	GO:0004872, GO:0005515, GO:0007160, GO:0007229, GO:0008305, GO:0016021

Table 2. Enriched data with background associations: First column lists "knowledge-poor" data in the form of GenBank identifiers, rows of second column show gene ontology terms at successive distances from the directly associated terms

AA458965	0: GO:0005125, GO:0005615, GO:0006955, GO:0007155
	1: GO:0005102, GO:0006952, GO:0007154, GO:0050874
	2: GO:0004871, GO:0005488, GO:0007582, GO:0009607, GO:0009987
	3: GO:0050896, 2 x GO:0003674, 2 x GO:0008150
	4: GO:0007582
	5: GO:0008150
AA490846	0: GO:0004872, GO:0005515, GO:0007160, GO:0007229, GO:0008305, GO:0016021
	1: GO:0004871, GO:0005488, GO:0007155, GO:0007166, GO:0043235
	2: 2 x GO:0003674, GO:0007154, GO:0007165, GO:0043234
	3: GO:0007154, GO:0005575, GO:0009987
	4: GO:0009987, GO:0008150
	5: GO:0008150

Note: Some terms are seen multiple times at the same distance or at further distances

background knowledge for the genes. As we trace back terms higher in the hierarchy we successively build up more general background knowledge for the genes. The complete set of associations for the genes in our example is shown in Table 2. It should be clear that the terms associated with the genes differ in importance. Terms that are lower in the hierarchy are more specific to the data items and should be treated as more significant for comparisons between data items. Conversely, terms that are far from the original terms (in terms of distance up the hierarchy) are more general and should play a less significant role in comparison of data items. Furthermore, different child terms may have the same parent term or terms. This means that as we trace back up the ontology hierarchy we may draw in the same term more than once. Consequently, the background knowledge of terms may, and usually will, have duplicated terms.

This observation suggests a method of applying a similar knowledge representation to that used in the field of computational linguistics. Rather than using a binary vector to represent the presence or absence of an association between data item and gene ontology term, we use a real value measure or weighting of the degree of significance of the term to the data item. Terms directly associated with each data item, for example those listed in Table 1, receive a weight value of 1, terms indirectly associated with the data item (i.e., higher in the hierarchy) are given a lower weighting and terms that cannot be reached from terms associated with the data item are assigned 0. This leads to a less sparse vector where comparisons may be made.

A straightforward method of deriving the distance between terms using a weighting scheme like this is to adapt a similarity measure called the Tanimoto measure (Theodoridis & Koutroumbas, 1999). The Tanimoto measure defines a measure of similarity between sets:

$$\frac{n_{X\cap Y}}{n_X + n_Y - n_{X\cap Y}} = \frac{n_{X\cap Y}}{n_{X\cup Y}}$$

where X and Y are the two sets being compared and n_X, n_Y and $n_{X\cap Y}$ are the number of elements in the sets X, Y and $X \cap Y$ respectively.

However, in the current situation, the "sets" being compared are the gene ontology terms for the two genes. As there may be duplicated terms in the lists associated with each data item we adapt the Tanimoto measure to give similarities between bags rather than sets.

Also, as the terms higher in the ontology are less significant in terms of comparison than the more specific terms towards the bottom, we weight the contribution of terms by the distance from the descendent gene ontology term directly associated with the gene. In effect, this results in a "weighted" cardinality of the bag of gene ontology terms. Furthermore, as we are interested in a distance rather than a similarity, we subtract the similarity from 1. The final distance function used, then, is:

$$D_{X,Y} = 1 - \frac{n'_{X\cap Y}}{n'_X + n'_Y - n'_{X\cap Y}} = 1 - \frac{n'_{X\cap Y}}{n'_{X\cup Y}}$$

where X and Y are the bags of terms being compared and n'_X, n'_Y and $n'_{X\cap Y}$ are the weighted cardinalities of the bags X, Y and $X \cap Y$ respectively given by:

$$n'_X = \sum_{i\in X} c^{d_i}$$

where X is the bag of gene ontology terms, d_i is the distance of term of X with index i from its associated descendent in the original set of gene ontology terms for the gene, and c is the weight constant. The weighted cardinality of the other bags is similarly defined.

The more general gene ontology terms provide a context for the understanding of the lower level terms directly associated with genes. The c weight constant allows variation of the importance of the "context" to the comparison. A value of $c = 0$ means that higher level are ignored. A value of 1 considers all terms equally irrespective of their position in the hierarchy and regards the very general terms as overly significant. The c parameter may be viewed as a sort of "constant of gravity" for the clusters. The higher the value of c, the more that distantly related genes are gathered into a cluster. A choice of $c = 0.9$ gives reasonable results.

A similar graph-based approach for determining similarity based on gene ontology relationships to our described above is given in Lee, Hur, and Kim (2004). That approach involves transformation of the gene ontology from a directed acyclic graph into a tree structure and encoding of gene ontology accession codes to map into the tree.

Our similarity function contains several assumptions about ontologies. It treats distances between levels in the ontology as the same. This means that terms that are the same distance away from the terms directly associated to data items have the same effect on the similarity measure. This may not necessarily reflect the knowledge encoded in the ontology. The level of fan-out from a parent to child in the ontology may be an indication of the concentration of knowledge in the ontology. For example, when the fan-out from parent to child is large, this may indicate that the parent concept has been investigated more or is understood better than parents with less fan-out. This and other measures could conceivably be incorporated into the similarity function.

Automatic Cluster Identification and Naming

Once clusters have been identified, the ontology can facilitate inference of cluster descriptions. The descriptions say how data items in the cluster are similar to one another and different to other clusters using the vocabulary of the ontology.

Cluster descriptions are inferred for each cluster using the method shown in the pseudo

code in Figure 4. At lines 1 and 2 the algorithm starts with an empty set of definitions and a list of all terms directly associated with the data items in the given cluster. The ontology hierarchy is traversed upwards replacing terms with their parent (more general) terms. Terms are replaced (line 12) only if the parent term is *not* associated with data items in another cluster (or is one of any of the ancestor terms in another cluster). At lines 8 and 14 the algorithm chooses a term to be added to the class description. Line 8 is the case when the top of the hierarchy is reached and line 14 is the case when no parent terms could be found that referred only to the cluster of interest. The output of the algorithm is a list of terms for a cluster that describe in the most general way possible the data items in the cluster (but not so general that it describes another cluster).

Insight into structure within clusters can be gained by examining which data items are associated with terms in the cluster description. It can happen that a subset of the data items in a cluster may have a description that is more concise than the description for all the data items in the cluster. This may be an indication of poor clustering of the data items.

CASE STUDY

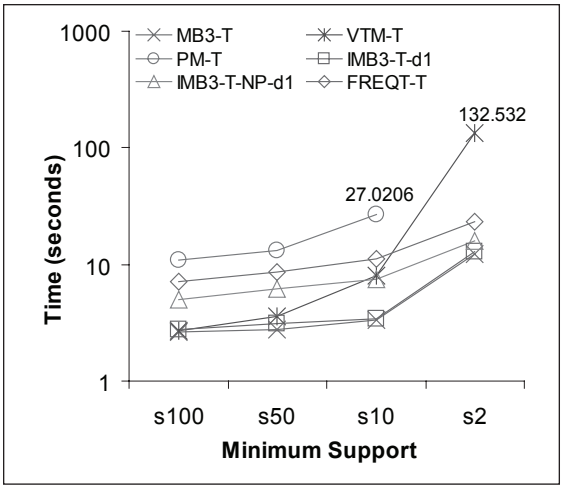
This section presents a case study of enriching bio-medical data with the protein ontology. The case study discusses the results of six data mining algorithms on PO data. The protein ontology database is created as an instance store for various protein data using the PO format. PO provides technical and scientific infrastructure to allow evidence based description and analysis of relationships between proteins. PO uses data sources like PDB, SCOP, OMIM and various published scientific literature to gather protein data. PO database is represented using OWL. PO database at the moment contains data instances of following protein families: (1) prion proteins, (2) B.Subtilis, (3) CLIC and (4) PTEN. More protein data instances will be added as PO is more developed. The PO instance store at moment covers various species of proteins from bacterial and plant proteins to human proteins. Such a generic representation using PO shows the strength of PO format representation.

We used some standard hierarchical and tree mining algorithms (Tan & Dillon, in press) on the PO database. We compared MB3-Miner

Figure 4. Pseudo code for cluster identification and naming

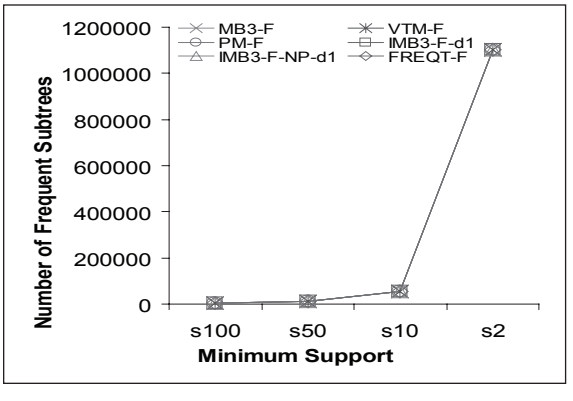
```
1   definitions = { }
2   working = terms directly associated with the data items in the cluster
3   while there are terms in working
4       new_working = { }
5       for each term in working
6           parents = parent terms of term
7           if there are no parents
8               add term to definitions
9           else
10              for each parent_term in parents
11                  if parent_term is associated only with this cluster
12                      add parent_term to new_working
13                  else
14                      add term to definitions
15      working = new_working
16  end while
17  definitions is the set of terms describing the cluster.
```

Figure 5. Time performance for prion dataset of PO data



(MB3), X3-Miner (X3), VTreeMiner (VTM) and PatternMatcher (PM) for mining embedded subtrees and IMB3-Miner (IMB3), FREQT (FT) for mining induced subtrees of PO data. In these experiments we are mining prion proteins dataset described using protein ontology framework, represented in OWL. For this dataset we map the OWL tags to integer indexes. The maximum height is 1. In this case all candidate subtrees generated by all algorithms would be induced subtrees. Figure 5 shows the time performance of different algorithms. Our original MB3 has the best time performance for this data.

Figure 6. Number of frequent subtrees for prion dataset of PO data



Quite interestingly, with prion dataset of PO the number of frequent candidate subtrees generated is identical for all algorithms (Figure 6). Another observation is that when support is less than 10, PM aborts and VTM performs poorly. The rationale for this could be because the utilized join approach enumerates additional invalid subtrees. Note that original MB3 is faster than IMB3 due to additional checks performed to restrict the level of embedding.

CONCLUSION

We discussed the two challenges to incorporate this biological knowledge into the data mining cycle: generating the ontologies, and adapting the data mining algorithms to make use of the ontologies. We present protein ontology (PO) framework, discuss semantic interoperability relationships between its concepts, and compare its structure with gene ontology (GO). We also demonstrate how to modify clustering algorithms in order to utilize the structure of GO. The results of six data mining algorithms on PO data are discussed, showing the strength of PO in enriching data for effective analysis.

REFERENCES

Diehn, M., Sherlock, G., Binkley, G., Jin, H., Matese, J. C., Hernandez-Boussard, T., et al. (2003). SOURCE: A unified genomic resource of functional annotations, ontologies, and gene expression data. *Nucleic Acids Research, 31*(1), 219-223.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (2nd ed). New York: John Wiley & Sons.

GO Consortium. (2001). Creating the gene ontology resource: Design and implementation. *Genome Research, 11*, 1425-1433.

GO Consortium & S. E. Lewis. (2004). Gene ontology: Looking backwards and forwards. *Genome Biology, 6*(1), 103.1-103.4.

Lee, S. G., Hur, J. U., & Kim, Y. S. (2004). A graph-theoretic modeling on GO space for biological interpretation of gene clusters. *Bioinformatics, 20*(3), 381-388.

Nelson, S. J., Schopen, M., et al. (2004, September 7-11).The MeSH translation maintenance system: Structure, interface design, and implementation. In M. Fieschi (Ed.), *Proceedings of the 11th World Congress on Medical Informatics*, San Francisco (pp. 67-69).

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge: Cambridge University Press.

Sidhu, A. S., & Dillon, T. S. (2006a). Protein ontology: Data integration using protein ontology. In Z. Ma & J. Y. Chen (Eds.), *Database modeling in biology: Practices and challenges*. New York: Springer Science, Inc.

Sidhu, A. S., & Dillon, T. S. (2006b). *Protein ontology project: 2006 Updates*. Invited paper presented at Data Mining and Information Engineering 2006, Prague, Czech Republic. WIT Press.

Sidhu, A. S., & Dillon, T. S. (2005a). Ontological foundation for protein data models. In *Proceedings of the First IFIP WG 2.12 & WG 12.4 International Workshop on Web Semantics (SWWS 2005) in conjunction with On The Move Federated Conferences (OTM 2005)*, Agia Napa, Cyprus (LNCS). Springer-Verlag.

Sidhu, A. S., & Dillon, T. S. (2005a). Protein ontology: Vocabulary for protein data. In *Proceedings of the 3rd IEEE International Conference on Information Technology and Applications (IEEE ICITA 2005)*, Sydney, Australia (pp. 465-469).

Tan, H., & Dillon. T.S. (in press). IMB3-miner: Mining induced/embedded subtrees by constraining the level of embedding. In *Proceedings of PAKDD 2006*.

Theodoridis, S., & Koutroumbas, K. (1999). *Pattern recognition*. San Diego: Academic Press.