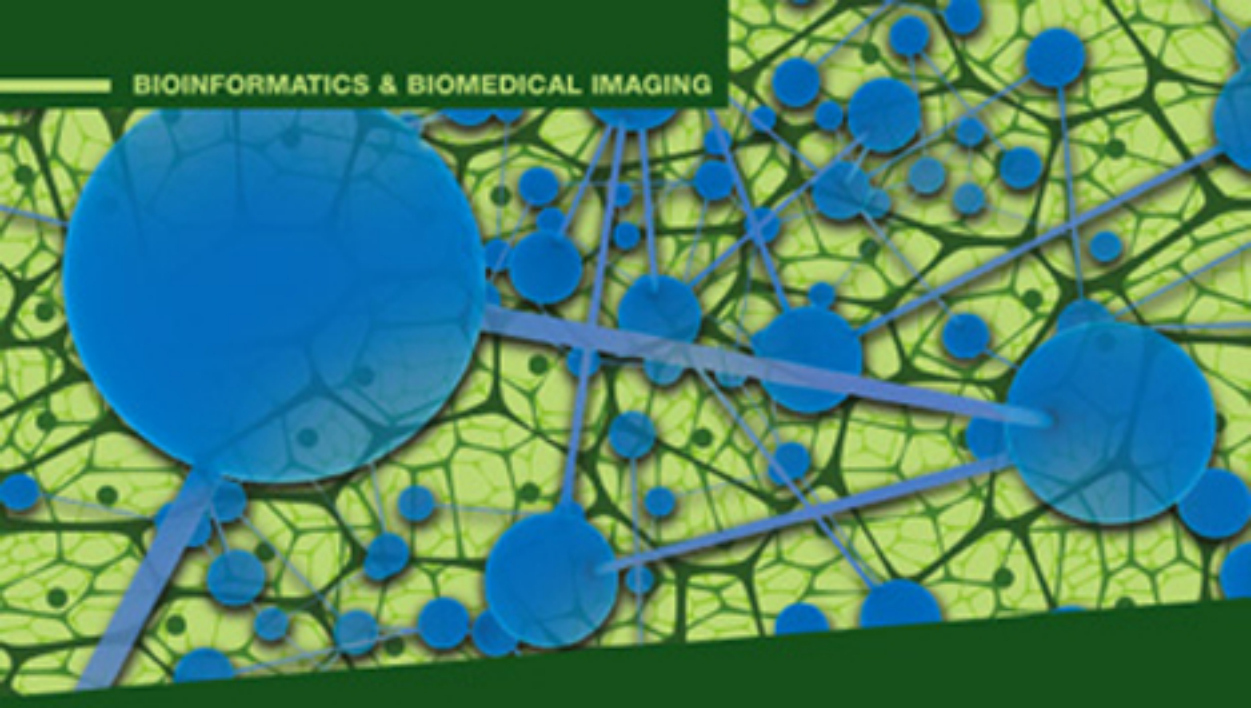
BIOINFORMATICS & BIOMEDICAL IMAGING



Data Mining in Biomedicine

Using Ontologies

Mihail Popescu • Dong Xu | editors

Data Mining in Biomedicine Using Ontologies

Artech House Series
Bioinformatics & Biomedical Imaging

Data Mining in Biomedicine Using Ontologies

Mihail Popescu
Dong Xu

Editors

ARTECH HOUSE
BOSTON | LONDON
artechhouse.com

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the U.S. Library of Congress.

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

ISBN-13: 978-1-59693-370-5

Cover design by Igor Valdman

685 Canton Street
Norwood, MA 02062

10 9 8 7 6 5 4 3 2 1

Contents

Foreword

Over the past decades, large amounts of biomedical data have become available, resulting in part from the "omics" revolution, that is, from the availability of high-throughput methods for analyzing biological structures (e.g., DNA and protein sequencing), as well as for running experiments (e.g., microarray technology for analyzing gene expression). Other large (and ever expanding) datasets include biomedical literature, available through PubMed/MEDLINE and, increasingly, through publicly available archives of full-text articles, such as PubMedCentral. Large clinical datasets extracted from electronic health records maintained by hospitals or the patient themselves are also available to researchers within the limits imposed by privacy regulations.

As is the case in other domains (e.g., finance or physics), data mining techniques have been developed or customized for exploiting the typically high-dimensional datasets of biomedicine. One prototypical example is the analysis and visualization of gene patterns in gene expression data, identified through clustering techniques, whose dendrograms and heat maps have become ubiquitous in the biomedical literature.

The availability of such datasets and tools for exploiting them has fostered the development of data-driven research, as opposed to the traditional hypothesis-driven research. Instead of collecting and analyzing data in an attempt to prove a hypothesis established beforehand, data-driven research focuses on the identification of patterns in datasets. Such patterns (and possible deviations from) can then suggest hypotheses and support knowledge discovery.

Biomedical ontologies, terminologies, and knowledge bases are artifacts created for representing biomedical entities (e.g., anatomical structures, genes), their names (e.g., *basal ganglia*, *dystrophin*), and knowledge about them (e.g., "the liver is contained in the abdominal cavity," "cystic fibrosis is caused by a mutation of the CFTR gene located on chromosome 7"). Uses of biomedical ontologies and related artifacts include knowledge management, data integration, and decision support. More generally, biomedical ontologies represent a valuable source of symbolic knowledge.

In several domains, the use of both symbolic knowledge and statistical knowledge has improved the performance of applications. This is the case, for example, in natural language processing. In biomedicine, ontologies are used increasingly in conjunction with data mining techniques, supporting data aggregation and semantic

normalization, as well as providing a source of domain knowledge. Here again, the analysis of gene expression data provides a typical example. In the traditional approach to analyzing microarray data, ontologies such as the Gene Ontology were used to make biological sense of the gene clusters obtained. More recent algorithms take advantage of ontologies as a source of prior knowledge, allowing this knowledge to influence the clustering process, together with the expression data.

The editors of this book have recognized the importance of combining data mining and ontologies for the analysis of biomedical datasets in applications, including the prediction of functional annotations, the creation of biological networks, and biomedical text mining. This book presents a wide collection of such applications, along with related algorithms and ontologies. Several applications illustrating the benefit of reasoning with biomedical ontologies are presented as well, making this book a rich resource for both computer scientists and biomedical researchers. The ontologist will see in this book the embodiment of *biomedical ontology in action*.

Olivier Bodenreider, Ph.D.
National Library of Medicine
August 2009

Preface

It has become almost a stereotype to start any biomedical data mining book with a statement related to the large amount of data generated in the last two decades as a motivation for the various solutions presented by the work in question. However, it is also important to note that the existing amount of biomedical data is still insufficient when describing the complex phenomena of life. From a technical perspective, we are dealing with a moving target. While we are adding multiple data points in a hypothetical feature space we are substantially increasing its dimension and making the problem less tractable. We believe that the main characteristic of the current biomedical data is, in fact, its diversity. There are not only many types of sequencers, microarrays, and spectrographs, but also many medical tests and imaging modalities that are used in studying life. All of these instruments produce huge amounts of very heterogeneous data. As a result, the real problem consists in integrating all of these data sets in order to obtain a deeper understanding of the object of study. In the meantime, traditional approaches where each data set was studied in its "silo" have substantial limitations. In this context, the use of ontologies has emerged as a possible solution for bridging the gap between silos.

An ontology is a set of vocabulary terms whose meanings and relations with other terms are explicitly stated. These controlled vocabulary terms act as adaptors to mitigate and integrate the heterogeneous data. A growing number of ontologies are being built and used for annotating data in biomedical research. Ontologies are frequently used in numerous ways including connecting different databases, refined searching, interpreting experimental/clinical data, and inferring knowledge.

The goal of this edited book is to introduce emerging developments and applications of bio-ontologies in data mining. The focus of this book is on the algorithms and methodologies rather than on the application domains themselves. This book explores not only how ontologies are employed in conjunction with traditional algorithms, but also how they transform the algorithms themselves. In this book, we denote the algorithms transformed by including an ontology component as ontological (e.g., ontological self-organizing maps). We tried to include examples of ontological algorithms as diversely as possible, covering description logic, probability, and fuzzy logic, hoping that interested researchers and graduate students will be able to find viable solutions for their problems. This book also attempts to cover major data-mining approaches: unsupervised learning (e.g., clustering and self-organizing maps), classification, and rule mining. However, we acknowledge that we left out many other related methods. Since this is a rapidly developing field that encompasses a very wide range of research topics, it is difficult

for any individual to write a comprehensive monograph on this subject. We are fortunate to be able to assemble a team of experts, who are actively doing research in bio-ontologies in data mining, to write this book.

Each chapter in this book is a self-contained review of a specific topic. Hence, a reader does not need to read through the chapters sequentially. However, readers not familiar with ontologies are suggested to read Chapter 1 first. In addition, for a better understanding of the probabilistic and fuzzy methods (Chapters 3, 5, 6, 7, 8, and 10) a previous reading of Chapter 2 is also advised. Cross-references are placed among chapters that, although not vital for understanding, may increase reader's awareness of the subject. Each chapter is designed to cover the following materials: the problem definition and a historical perspective; mathematical or computational formulation of the problem; computational methods and algorithms; performance results; and the strengths, pitfalls, challenges, and future research directions.

A brief description of each chapter is given below.

Chapter 1 (Introduction to Ontologies) provides definition, classification, and a historical perspective on ontologies. A review of some applications, tools, and a description of most used ontologies, GO and UMLS, are also included.

Chapter 2 (Ontological Similarity Measures) presents an introduction together with a historic perspective on object similarity. Various measures of ontology term similarity (information content, path based, depth based, etc.), together with most used object-similarity measures (linear order statistics, fuzzy measures, etc.) are described. Some of these measures are used in the approximate reasoning examples presented in the following chapters.

Chapter 3 (Clustering with Ontologies) introduces several relational clustering algorithms that act on dissimilarity matrices such as non-Euclidean relational fuzzy C-means and correlation cluster validity. An ontological version of self-organizing maps is also described. Examples of applications of these algorithms on some test data sets are also included.

Chapter 4 (Analyzing and Classifying Protein Family Data Using OWL Reasoning) describes a method for protein classification that uses ontologies in a description logic framework. The approach is an example of emerging algorithms that combine database technology with description logic reasoning.

Chapter 5 (GO-based Gene Function and Network Characterization) describes a GO-based probabilistic framework for gene function inference and regulatory network characterization. Aside from using ontologies, the framework is also relevant for its integration approach to heterogeneous data in general.

Chapter 6 (Mapping Genes to Biological Pathways Using Ontological Fuzzy Rule Systems) provides an introduction to ontological fuzzy rule systems. A brief introduction to fuzzy rule systems is included. An application of ontological fuzzy rule systems to mapping genes to biological pathways is also discussed.

Chapter 7 (Extracting Biological Knowledge by Fuzzy Association Rule Mining) describes a fuzzy ontological extension of association rule mining, which is possibly the most popular data-mining algorithm. The algorithm is applied to extracting knowledge from multiple microarray data sources.

Chapter 8 (Data Summarization Using Ontologies) presents another approach to approximate reasoning using ontologies. The approach is used for creat-

ing conceptual summaries using a connectivity clustering method based on term similarity.

Chapter 9 (Reasoning over Anatomical Ontologies) presents an ample review of reasoning with ontologies in bioinformatics. An example of ontological reasoning applied to maize tassel is included.

Chapter 10 (Ontology Application in Text Mining) presents an ontological extension of the well-known Swanson's Undiscovered Public Knowledge method. Each document is represented as a graph (network) of ontology terms. A method for clustering scale-free networks nodes is also described.

We have selected these topics carefully so that the book would be useful to a broad readership, including students, postdoctoral fellows, professional practitioners, as well as bioinformatics/medical informatics experts. We expect that the book can be used as a textbook for upper undergraduate-level or beginning graduate-level bioinformatics/medical informatics courses.

Mihail Popescu
Assistant professor of medical informatics,
University of Missouri

Dong Xu
Professor and chair, Department of Computer Science,
University of Missouri
August 2009

CHAPTER 1

Introduction to Ontologies

Andrew Gibson and Robert Stevens

There have been many attempts to provide an accurate and useful definition for the term ontology, but it remains difficult to converge on one that covers all of the modern uses of the term. So, when first attempting to understand modern ontologies, a key thing to remember is to expect diversity and no simple answers. This chapter aims to give a broad overview of the different perspectives that give rise to the diversity of ontologies, with emphasis on the different problems to which ontologies have been applied in biomedicine.

1.1 Introduction

We say that we know things all the time. I know that this is a book chapter, and that chapters are part of books. I know that the book will contain other chapters, because I have never seen a book with only one chapter. I do know, though, that it is possible to have books without a chapter structure. I know that books are found in libraries and that they can be used to communicate teaching material.

I can say all of the things above without actually having to observe specific books, because I am able to make abstractions about the world. As we observe the world, we start to make generalizations that allow us to refer to types of things that we have observed. Perhaps what I wrote above seems obvious, but that is because we share a view of the world in which these concepts hold a common meaning. This shared view allows me to communicate without direct reference to any specific book, library, or teaching and learning process. I am also able to communicate these concepts effectively, because I know the terms with which to refer to the concepts that you, the reader, and I, the writer, both use in the English language.

Collectively, concepts, how they are related, and their terms of reference form *knowledge*. Knowledge can be expressed in many ways, but usually in natural language in the form of speech or text. Natural language is versatile and expressive, and these qualities often make it ambiguous, as there are many ways of communicating the same knowledge. Sometimes there are many terms that have the same or similar meanings, and sometimes one term can have multiple meanings that need to be clarified through the context of their use. Natural language is the standard form of communicating about biology.

Ontologies are a way of representing knowledge in the age of modern computing [1]. In an ontology, a vocabulary of terms is combined with statements about the relationships among the entities to which the vocabulary refers. The ambiguous structure of natural language is replaced by a structure from which the same meaning can be consistently accessed computationally. Ontologies are particularly useful for representing knowledge in domains in which specialist vocabularies exist as extensions to the common vocabulary of a language.

Modern biomedicine incorporates knowledge from a diverse set of fields, including chemistry, physics, mathematics, engineering, informatics, statistics, and of course, biology and its various subdisciplines. Each one of these disciplines has a large amount of specialist knowledge. No one person can have the expertise to know it all, and so we turn to computers to make it easier to specify, integrate, and structure our knowledge with ontologies.

1.2 History of Ontologies in Biomedicine

In recent years, ontologies have become more visible within bioinformatics [1], and this often leads to the assumption that such knowledge representation is a recent development. In fact, there is a large corpus of knowledge-representation experience, especially in the medical domain, and much of it is still relevant today. In this section, we give an overview of the most prominent historical aspects of ontologies and the underlying developments in knowledge representation, with a specific focus on biomedicine.

1.2.1 The Philosophical Connection

Like *biology,* the word *ontology* is conventionally an uncountable noun that represents the *field of ontology.* The term *an ontology,* using the indefinite article and suggesting that more than one ontology exists, is a recent usage of the word that is now relatively common in informatics disciplines. This form has not entered mainstream language and is not yet recognized by most English dictionaries. Standard reference definitions reflect this: "Ontology. Noun: Philosophy: The branch of metaphysics concerned with the nature of being" [2].

The philosophical field of ontology can be traced back to the ancient Greek philosophers [3], and it concerns the categorization of existence at a very fundamental and abstract level. As we will see, the process of building ontologies also involves categorization. The terminological connection between *ontology* and *ontologies* has produced a strong link between the specification of knowledge-representation schemes for information systems and the philosophical exercise of partitioning existence.

1.2.2 Recent Definition in Computer Science

The modern use of the term ontology emerged in the early 1990s from research into the specification of knowledge as a distinct component of knowledge-based systems in the field of artificial intelligence (AI). Earlier attempts at applying AI techniques

in medicine can be found in expert systems in the 1970s and 1980s [4]. The idea of these systems was that a medical expert could feed information on a specific medical case into a computer programmed with detailed background medical knowledge and then receive advice from the computer on the most likely course of action. One major problem was that the specification of expert knowledge for an AI system represents a significant investment in time and effort, yet the knowledge was not specified in a way that could be easily reused or connected across systems.

The requirement for explicit ontologies emerged from the conclusion that knowledge should be specified independently from a specific AI application. In this way, knowledge of a domain could be explicitly stated and shared across different computer applications. The first use of the term in the literature often is attributed to Thomas Gruber [5], who provides a description of ontologies as components of knowledge bases: "Vocabularies or representational *terms*—classes, relations, functions, object constants—with agreed-upon *definitions*, in the form of human readable text and machine enforceable, declarative constraints on their well formed use" [5].

This description by Gruber remains a good description of what constitutes an ontology in AI, although, as we will see, some of the requirements in this definition have been relaxed as the term has been reused in other domains. Gruber's most-cited article [6] goes on to abridge the description into the most commonly quoted concise definition of an ontology: "An ontology is an explicit specification of a conceptualization."

Outside of the context of this article, this definition is not very informative and assumes an understanding of the context and definition of both *specification* and *conceptualization*. Many also find this short definition too abstract, as it is unclear what someone means when he or she says, "I have built an ontology." In many cases, it simply means an encoding of knowledge for computational purposes. Definition aside, what Gruber had identified was a clear challenge for the engineering of AI applications. Interestingly, Gruber also denied the connection between ontology in informatics and ontology in philosophy, though, in practice, the former is at least often informed by the latter.

1.2.3 Origins of Bio-Ontologies

The term ontology appears early on in the publication history of bioinformatics. The use of an ontology as a means to give a high-fidelity schema of the E. coli genome and metabolism was a primary motivation for its use in the EcoCyc database [7, 8]. Systems such as TAMBIS [9] also used an ontology as a schema (see Section 1.6.6). Karp [10] advocated ontologies as means of addressing the severe heterogeneity of description in biology and bioinformatics and the ontology for molecular biology [11] was an early attempt in this direction. This early use of ontologies within bioinformatics was also driven from a computer-science perspective.

The widespread use of the term *ontology* in biomedicine really began in the 2000, when a consortium of groups from three major model-organism databases announced the release of the Gene Ontology (GO) database [12]. Since then, GO has been highly successful and has prompted many more bio-ontologies to follow the aim of unifying the vocabularies of over 60 distinct domains of biology, such

as cell types, phenotypic and anatomical descriptions of various organisms, and biological sequence features. These vocabularies are all developed in coordination under the umbrella organization of the Open Biomedical Ontologies (OBO) Consortium [13]. GO is discussed in more detail in Section 1.5.1.

This controlled-vocabulary form of ontology evolved independently of research from the idea of ontologies in the AI domain. As a result, there are differences in the way in which the two forms are developed, applied, and evaluated. Bio-ontologies have broadened the original meaning of ontology from Gruber's description to cover knowledge artifacts that have the primary function of a controlled structured vocabulary or terminology. Most bio-ontologies are for the annotation of data and are largely intended for human interpretation, rather than computational inference [1], meaning that most of the effort goes into the consistent development of an agreed-upon terminology. Such ontologies do not necessarily have the "machine enforceable, declarative constraints" of Gruber's description of the ontology that would be essential for an AI system.

1.2.4 Clinical and Medical Terminologies

The broadening of the meaning of ontology has resulted in the frequent and sometimes controversial inclusion of medical terminologies as ontologies. Medicine has had the problem of integrating and annotating data for centuries [1], and controlled vocabularies can be dated back to the 17th century in the London Bills of Mortality [60]. One of the major medical terminologies of today is the International Classification of Diseases (ICD) [61], which is used to classify mortality statistics from around the world. The first version of the ICD dates back to the 1880s, long before any computational challenges existed. The advancement and expansion of clinical knowledge predates the challenges addressed by the OBO consortium by some time, but the principles were the same. As a result, a diverse set of terminologies were developed that describe particular aspects of medicine, including anatomy, physiology, diseases and disorders, symptoms, diagnostics, treatments, and protocols. Most of these have appeared over the last 30 years, as digital information systems have become more ubiquitous in healthcare environments. Unlike the OBO vocabularies, however, many medical terminologies have been developed without any coordination with other terminologies. The result is a lot of redundancy and inconsistency across vocabularies [14]. One of the major challenges in this field today is the harmonization of terminologies [15].

1.2.5 Recent Advances in Computer Science

Through the 1990s, foundational research on ontologies in AI became more prominent, and several different languages for expressing ontologies appeared, based on several different knowledge-representation paradigms [16]. In 2001, a vision for an extension to the Web—the Semantic Web—was laid out to capture computer-interpretable data, as well as content for humans [17, 18]. Included in this vision was the need for an ontology language for the Web. A group was set up by the World Wide Web Consortium (W3C) that would build on and extend some of the earlier ontology languages to produce an internationally recognized language standard. The

knowledge-representation paradigm chosen for this language was description logics (DL) [19]. The first iteration of this standard—the Web Ontology Language (OWL) [20]—was officially released in 2004. Very recently, a second iteration (OWL2) was released to extend the original specification with more features derived from experiences in using OWL and advances in automated reasoning.

Today, OWL and the Resource Description Framework (RDF), another W3C Semantic Web standard, present a means to achieve integration and perform computational inferencing over data. Of particular interest to biomedicine, the ability of Web ontologies to specify a global schema for data supports the challenge of data integration, which remains one of the primary challenges in biomedical informatics. Also appealing to biomedicine is the idea that, given an axiomatically rich ontology describing a particular domain combined with a particular set of facts, a DL reasoner is capable of filling in important facts that may have been overlooked or omitted by a researcher, and it may even generate a totally new discovery or hypothesis [21].

1.3 Form and Function of Ontologies

This section aims to briefly introduce some important distinctions in the content of ontologies. We make a distinction between the form and function of an ontology. In computer files, the various components of ontologies need to be specified by a syntax, and this is their form. The function of an ontology depends on two aspects: the combination of ontology components used to express the encoded knowledge in the ontology, and the style of representation of the knowledge. Different ontologies have different goals, which in turn, require particular combinations of ontology components. The resulting function adds a layer of meaning onto the form that allows it to be interpreted by humans and/or computers.

1.3.1 Basic Components of Ontologies

All ontologies have two necessary components: entities and relationships [22]. These are the main components that are necessarily expressed in the form of the ontology, with the relationships between the entities providing the structure for the ontology.

The entities that form the nodes of an ontology are most commonly referred to as *concepts* or *classes*. Less common terms for these are *universals*, *kinds*, *types,* or *categories*, although their use in the context of ontologies is discouraged because of connotations from other classification systems. The relationships in an ontology are most commonly known as *properties*, *relations,* or *roles*. They are also sometimes referred to as *attributes*, but this term has meaning in other knowledge-representation systems, and it is discouraged. Relationships are used to make statements that specify associations between entities in the ontology. In the form of the ontology, it is usually important that each of the entities and relationships have a unique identifier.

Most generally, a combination of entities and relationships (nodes and edges) can be considered as a directed acyclic graph; however, the overall structure of an

ontology is usually presented as a hierarchy that is established by linking classes with relationships, going from more general to more specific. Every class in the hierarchy of an ontology will be related to at least one other class with one of these relationships. This structure provides some general root or top classes (e.g., *cell*) and some more specific classes that appear further down the hierarchy (e.g., *tracheal epithelial cell*). The relations used in the hierarchy are dependant on the function of the ontology. The most common hierarchy-forming relationship is the *is a* relationship (e.g., *tracheal epithelial cell is an epithelial cell*). Another common hierarchy-forming relationship is *part of*, and ontologies that only use part of in the hierarchy are referred to as *partonomies*. In biomedicine, partonomies are usually associated with ontologies of anatomical features, where a general node would be *human body*, with more specific classes, such as *arm*, *hand*, *finger*, and so on.

1.3.2 Components for Humans, Components for Computers

The form of the ontology exists primarily so that the components can be computationally identified and processed. Ontologies, however, need to have some sort of meaning [23]. In addition to the core components, there are various additional components that can contribute to the function of an ontology.

First, to help understand what makes something understandable to a computer, consider the following comparison with programming languages. A precise syntax specification allows the computer, through the use of a compiler program, to correctly interpret the intended function of the code. The syntax enables the program to be parsed and the components determined. The semantics of the language allow those components to be interpreted correctly by the compiler; that is, what the statements mean. As in programming, there are constructs available, which can be applied to entities in an ontology, that allow additional meaning to be structured in a way that the computer can interpret. In addition, a feature of good computer code will be in-line comments from the programmer. These are "commented out" and are ignored by the computer when the program is compiled, but are considered essential for the future interpretation of the code by a programmer.

Ontologies also need to make sense to humans, so that the meaning encoded in the ontology can be communicated. To the computer, the terms used to refer to classes mean nothing at all, and so they can be regarded as for human benefit and reference. Sometimes this is not enough to guarantee human comprehension, and more components can be added that annotate entities to further illustrate their meaning and context, such as comments or definitions. These annotations are expressed in natural language, so they also have no meaning for the computer. Ontologies can also have metadata components associated with them, as it is important to understand who wrote the ontology, who made changes, and why.

State-of-the-art logic-based languages from the field of AI provide powerful components for ontologies that add computational meaning (semantics) to encoded knowledge [23]. These components build on the classes and relationships in an ontology to more explicitly state what is known in a computationally accessible way. Instead of a compiler, ontologies are interpreted by computers through the use of a reasoner [19]. The reasoner can be used to check that the asserted facts in the ontology do not contradict one another (the ontology is consistent), and it can use

the encoded meaning in the ontology to identify facts that were not explicitly stated in the original ontology (computational inferences). An ontology designer has to be familiar with the implications of applying these sorts of components if they are to make the most of computational reasoning, which requires some expertise and appreciation for the underlying logical principles.

1.3.3 Ontology Engineering

The function of an ontology always requires that the knowledge is expressed in a sensible way, whether that function is for humans to be able to understand the terminology of a domain or for computers to make inferences about a certain kind of data. The wider understanding of such stylistic ontology engineering as a general art is at an early stage, but most descriptions draw an analogy with software engineering [24]. Where community development is carried out, it has been necessary to have clear guidelines and strategies for the naming of entities (see, for instance, the GO style guide at http://www.geneontology.org) [25]. Where logical formalisms are involved for computer interpretation of the ontology, raw expert knowledge sometimes needs to be processed into a representation of the knowledge that suits the particular language, as most have limitations on what sort of facts can be accurately expressed computationally. Ontologies are also influenced often by philosophical considerations, which can provide extra criteria for the way in which knowledge is encoded in an ontology. This introduction is not the place for a review of ontology-building methodologies, but Corcho, et al., [16] provides a good summary of approaches. The experiences of the GO are also illuminating [25].

1.4 Encoding Ontologies

The process of ontology building includes many steps, from scoping to evaluation and publishing, but a central step is encoding the ontology itself. OWL and the OBO format are two key knowledge-representation styles that are relevant to this book. As it is crucial for the development and deployment of ontologies that effective tool support is also provided, we will also review aspects of the most prominent open-source tools.

1.4.1 The OBO Format and the OBO Consortium

Most of the bio-ontologies developed under the OBO consortium are developed and deployed in OBO format. The format has several primary aims, the most important being human readability and ease of parsing. Standard data formats, such as XML, were not designed to be read by humans, but in the bioinformatics domain, this is often deemed necessary. Also in bioinformatics, such files are commonly parsed with custom scripts and regular expressions. XML format would make this difficult, even though parsers are automatically generated from XML schema. OBO format also has the stated aims of extensibility and minimal redundancy. The key structure in an OBO file is the stanza. These structures represent the components of the OBO file. Here is an example of a term stanza from the cell-type ontology:

```
[Term]
id: CL:0000058
name: chondroblast
is_a: CL:0000055 ! non-terminally differentiated cell
is_a: CL:0000548 ! animal cell
relationship: develops_from CL:0000134 ! mesenchymal cell
relationship: develops_from CL:0000222 ! mesodermal cell
```

Each OBO term stanza begins with an identifier tag that uniquely identifies the term and a name for that term. Both of these are required tags for any stanza in the OBO format specification. Following that are additional lines in the stanza that further specify the features of the term and relate the current term to other terms in the ontology through various relationships. The full specification of the OBO syntax is available from the Gene Ontology Web site (http://www.geneontology.org/).

One of the strongest points about OBO ontologies is their coordinated and community-driven approach. OBO ontologies produced by following the OBO consortium guidelines try to guarantee the uniqueness of terms across the ontologies. Each term is assigned a unique identifier, and each ontology is assigned a unique namespace. Efforts to reduce the redundancy of terms across all of the ontologies are ongoing [13]. Identifiers are guaranteed to persist over time, through a system of deprecation that manages changes in the ontology as knowledge evolves. This means that if a particular term is superseded, then that term will persist in the ontology, but will be flagged as obsolete. The OBO process properly captures the notion of separation between the concept (class, category, or type) and the label or term used in its rendering. It would be possible to change *glucose metabolic process* to *metabolism of glucose* without changing the underlying conceptualization; thus in this case, the identifier (GO:0006006) stays the same. Only when the underlying definition or conceptualization changes are new identifiers introduced for existing concepts. Many ontologies operate naming conventions through the use of singular nouns for class names; use of all lower case or initial capitals; avoidance of acronyms; avoidance of characters, such as -, /, !, and avoidance of with, of, and, and or.

Such stylistic conventions are a necessary parts of ontology construction; however, for concept labels, all the semantics or meaning is bound up within the natural-language string. As mentioned earlier in Section 1.3.2, this is less computationally accessible to a reasoner, although it is possible to extract some amount of meaning from consistently structured terms.

Recently, a lot of attention has been focussed on understanding how the statements in OBO ontologies relate to OWL. A mapping has been produced, so that the OBO format can be considered as an OWL syntax [26, 27]. It is worth noting that each OBO term is (philosophically) considered to be a class, the instances of which are entities in the real world. As such, the mapping to OWL specifies that OBO terms are equivalent to OWL classes (though an OWL class would not have the stricture of corresponding to a real-world entity, but to merely have instances).

1.4.2 OBO-Edit—The Open Biomedical Ontologies Editor

The main editor used for OBO is OBO-Edit [28]. This is an open source tool, and it has been designed to support the construction and maintenance of bio-ontologies in the OBO format. OBO-Edit has evolved along with the format to match the needs of those building and maintaining OBO, and it has benefited from the direct feedback of the community of OBO developers.

The user interface of OBO-Edit features an ontology editor panel that contains hierarchies of both the classes, relations, and obsolete terms in the ontology, which can be browsed with simple navigation. The hierarchy of classes supports the use of multiple relationships to create the backbone of the hierarchy. For example, the relation *develops from* can be used to create a visual step in the hierarchy, and where it is used, it will be indicated with a specific symbol. This is a convenient visual representation of the relationships in the ontology, and it helps with browsing.

The interface is also strongly oriented toward the tasks of search and annotation. OBO, like GO, are large, and finding terms is essential for the task of annotating genes. Many classes include a natural-language definition, comments, synonyms, and cross-references to other databases, and features for editing these fields are prominent in the interface. While an OBO mapping to OWL is available, by contrast the OBO-Edit interface has limited support for the specification of computer-interpretable ontology components.

1.4.3 OWL and RDF/XML

Web Ontology Language [20] is a state-of-the-art ontology language that includes a set of components that allow specific statements about classes and relations to be expressed in ontologies. These components have a well-defined (computer-interpretable) semantics, and therefore, the function of OWL can be strongly oriented toward computer-based interpretation of ontologies. A subset of OWL has been specified (OWL-DL) that includes only ontology statements that are interpretable by a DL reasoner. As described previously, this means the ontology can be checked for consistency, and computational inferences can be made from the asserted facts. OWL is flexible, and it is possible to represent artifacts, such as controlled vocabularies, complete with human-readable components, such as comments and annotations. Another often-cited advantage of OWL is its interoperability, because it can also be used as a data-exchange format.

In its form, OWL can be encoded in a number of recognized syntaxes, the most common being RDF/XML. This format is not meant to be human readable, but the Manchester syntax has been designed to be more human readable to address this issue [29]. As OWL is designed for the Web, any OWL ontology or component of an OWL ontology is assigned a Unique Resource Identifier (URI). It is not possible to adequately describe the sorts of statements that OWL-DL supports in this chapter. To support this topic, we recommend working through the "pizza tutorial" that has been designed for this purpose, as real understanding comes through experience, rather than a brief explanation (see http://www.co-ode.org) [30].

In terms of development, it is important to identify the intended function of an OWL ontology. For ontologies that do not require much of the expressive power

of OWL, the main difference is in the tool support for this task. When, however, OWL ontology development starts to include many of the more specialized features of OWL to make use of computational reasoning, then development starts to require more specialized developers who understand both the semantics of the language and the knowledge from the domain that they are encoding. When medical expert systems were being developed, a person in this role was known as a knowledge engineer, a role which is also relevant today. Community maintenance of a highly expressive ontology is more challenging than community development of controlled vocabularies, as the community has to both understand and agree on the logical meaning, as well as on the terms and natural-language definitions being used. In this sense, OWL ontology development has no clear community of practice in biomedicine, as the OBO community does.

1.4.4 Protégé—An OWL Ontology Editor

There are a number of editors and browsers available for OWL ontologies. Here, we focus on the Protégé ontology editor, as it is freely available, open source, and the focus of important OWL tutorial material.

OWL ontologies can become very large and complicated knowledge representations. As OWL became the de facto standard for ontology representation on the Web, the Protégé OWL ontology editor was adapted from earlier knowledge-representation languages to provide support for the development of such ontologies. The focus in development has been to provide the user with access to all of the components of OWL that make it possible to specify computationally interpretable ontologies. The user interface focuses heavily on the specification of such OWL components. The user will benefit most from this if he or she has or expects to gain a good working knowledge of the implications of the logical statements made in the ontology. Of course, the user does not have to use all of the expressivity of OWL, and the Protégé interface is also well suited to the development of simpler class hierarchies. The interface does not cater to the specific needs of any particular subcommunity of ontology developers; however, the most recent implementation of Protégé (Protégé 4—http://protege.stanford.edu) features a fully customizable interface that can be tailored to the preferences of the user and that is also designed so that plug-in modules can be developed easily for specific needs. Protégé 4 also allows OBO ontologies to be opened and saved directly, and it supports a number of other syntaxes. Importantly, recent versions of Protégé include fully integrated access to OWL DL reasoners, which means users can now easily benefit from computational inference.

1.5 Spotlight on GO and UMLS

Within the domain of biology and medicine, the two resources of GO and UMLS have arguably had the greatest impact [31]. As an introduction to where they are referenced in the other chapters in this book, we put a spotlight on the key aspects of these resources.

1.5.1 The Gene Ontology

At the time of its conception, the need for GO was powerful and straightforward: different molecular-biology databases were using different terms to describe important information about gene products. This heterogeneity was a barrier to the integration of the data held in these databases. The desire for such integration was driven by the advent of the first model organism genome sequences, which provided the possibility of performing large-scale comparative genomic studies. GO was revolutionary within bioinformatics because it provided a controlled vocabulary that could be used to annotate database entries. After a significant amount of investment and success, GO is now widely used. The usage of GO has expanded since its use for the three original genome database members of the consortium, and it has now been adopted by over 40 species-specific databases. Of particular note is the Gene Ontology Annotation (GOA) Project, which aims to ensure widespread annotation of UniProtKB entries with GO annotations [32]. This resource currently contains over 32 million GO annotations to more than 4.3 million proteins, through a combination of manual and automatic annotation methods.

GO is actually a set of three distinct vocabularies containing terms that describe three important aspects of gene products. The molecular function vocabulary includes terms that are used to describe the various elemental activities of a gene product. The biological process vocabulary includes terms that are used to describe the broader biological processes in which gene products can be involved and that are usually achieved through a combination of molecular functions. Finally, the cellular component vocabulary contains terms that describe the various locations in a cell with which gene products may be associated. For example, a gene product that acts as a transcription factor involved in cell cycle regulation may be annotated with the molecular functions of *DNA binding* and *transcription factor activity*, the biological processes of *transcription* and *G1/S transition of mitotic cell* cycle, and the cellular location of *nucleus*. In this case, these terms are independent of species, and so gene products annotated with these terms could be extracted from many different species-specific databases to facilitate comparative analysis in an investigation into cell cycle regulation. GO does contain terms that are not applicable to all species, but these are derived from the need for terms that describe aspects that are particular to some organisms; for example, no human gene products would be annotated with *cell wall*.

The process of annotating a gene product is the specification of an assertion about that gene product. Because of this, GO annotations cannot be made without some sort of evidence as to the source of the assertion. For this, GO also has evidence codes that can be associated with any annotation. There are two broad categories of evidence codes that distinguish between whether the annotation was made based on evidence that was derived from direct experimentation, such as a laboratory assay, or whether it was from indirect evidence, such as a computational experiment or a statement by an author in which the evidence is unclear. Annotations should always include citations of their sources. When annotations are being used for data mining, the type of evidence can be an important discriminatory factor.

As of March 2009, the GO Web site states that GO includes more than 16,000 biological process terms, over 2,300 cellular component terms, and over 8,500 molecular function terms. The curation (i.e., term validation) process means that almost all of these terms have a human-readable definition, which is important for getting more accurate annotations from the process. These terms may also have other relevant information, such as synonymous terms and cross references to other databases.

As the number of databases and data from different species and biological domains increases, so does the demand for more specific terms with which gene products can be annotated. The GO consortium organizes interest groups for specific domains that are intended to extend and improve the terms in the ontology. The terms in the ontologies are curated by a dedicated team, but requests for modifications and improvements can be requested by anybody, and so there is a strong sense of community development. The style of terms in the gene ontology is highly consistent [33]. Nearly all of the terms in the GO biological process to do with metabolism of chemicals follow the structure "<chemical> metabolism | biosynthesis | catabolism." Such a structure aids both the readability and the computational manipulation of the set of labels in the ontology [33, 34].

In data mining, GO is now widely used in a variety of ways to provide a functional perspective on the analysis of molecular biological data. The analysis of microarray results through analyzing the over-representation of GO terms within the differentially represented genes (e.g., [35, 36]) is a common usage. Other important examples include the functional interpretation of gene expression data and the prediction of gene function through similarity. The controlled vocabulary specified by GO also has useful applications in text mining. Specific examples of these and other uses are detailed in Chapters 5, 6, and 7.

1.5.2 The Unified Medical Language System

As mentioned previously, many biomedical vocabularies have evolved independently and have had virtually no coordinated development. This has led to much overlap and incompatibility between them, and integrating them is a significant challenge. The Unified Medical Language System (UMLS) addresses this challenge, and has been a repository of biomedical vocabularies developed by the U.S. National Library of Medicine for over 20 years [37, 38]. UMLS comprises three knowledge sources: the UMLS Metathesaurus, the Semantic Network, and the SPECIALIST Lexicon. Together, they seek to provide a set of resources that can aid in the analysis of text in the biomedical domain, from health records to research papers. By coordinating a wide range of vocabularies with lexical information, UMLS seeks to provide a language-oriented knowledge resource.

The Metathesaurus integrates over 100 vocabularies from a diverse set of biomedical fields, including diagnoses, procedures, clinical observations, signs and symptoms, drugs, diseases, anatomy, and genes. Notable resources include SNOMed-CT, GO, MeSH, NCI Thesaurus, OMIM, HL7, and ICD. The Metathesaurus is a set of biomedical and health-related concepts that are referred to by this diverse set of vocabularies, using different terms. A UMLS concept is something in biomedicine that has common meaning [39]. UMLS does not seek to develop its own

ontology that covers the domain of biomedicine, but instead provides a mapping between existing ontologies and terminologies. The result is an extensive set of more than 1 million concepts and 4 million terms for those concepts.

Each concept in the Metathesaurus is associated with a number of synonymous terms collected from the integrated vocabularies and has its own concept identifier that is unique in the Metathesaurus. The UMLS has a system for specifying information about the terms that it integrates, which provides other identifiers for atoms (for each term from each vocabulary), strings (for the precise lexical structure of a term, such as the part of speech, singular, and plural forms of a term), and terms (for integrating lexical variants of a term, such as haemoglobin and hemoglobin). Concepts integrate terms, strings, and atoms, so that as much of the information about the original terminology is preserved as possible. The SPECIALIST Lexicon stores more information on parts of speech and spelling variants for terms within UMLS, as well as common English words and other terminology found within medical literature.

Concepts in the Metathesaurus are linked to each other by relationships that either have been generated from the source vocabularies or have been specified during curation. Every concept in the Metathesaurus is also linked to the third major component of UMLS—the Semantic Network. This is essentially a general ontology for biomedicine that contains 135 semantic, hierarchically organized types and 54 types of relationships that can exist between these types [40]. Every concept in the Metathesaurus is assigned at least one semantic type.

As the UMLS is an integration of knowledge from many different resources, it inherits the gaps and shortcomings of the vocabularies that it integrates. This has not, however, prevented the extensive application of the UMLS, in particular, within text mining [41]. The Metathesaurus, Semantic Network, and SPECIALIST Lexicon together form a powerful set of resources for manipulating text. For example, there are several programs available within UMLS for marking up text, such as abstracts, with terms found from within UMLS (MetaMap, a tool for providing spelling variants for terms found within a text that facilitates parsing (lvg)), and customizing UMLS to provide the vocabularies needed for a particular task (MetamorphoSys). There are many examples of UMLS being used in text mining within bioinformatics. Some examples are the annotation of enzyme classes [42], the study of single nucleotide polymorphisms [43], and the annotation of transcription factors [44].

1.6 Types and Examples of Ontologies

In this chapter, we have looked at the historical development of ontologies, their components, representations, and engineering. Their uses have been illustrated along the way, but in this section, we will take a longer look at the different types of knowledge artifacts that are referred to as ontologies. Ideally, it would be simple to accurately classify the types of ontologies featured in this section. It can be surprising that in a field that is concerned with the classification of things, that there is no agreed-upon classification of ontologies themselves, even though there are clear differences. Part of the reason for this is simply a lack of a sufficiently rich vocabulary

for talking about ontologies. There are, however, some broad classifications of ontologies that have diverse uses, and any one ontology or ontologylike artifact can fall into one or more of the categories described below. This list is not exhaustive, but the most prominent examples are highlighted.

1.6.1 Upper Ontologies

Upper ontologies are often referred to as top ontologies or foundational ontologies. They strongly reflect the philosophical roots of ontological classification. They do not cover any specific domain or application, and instead make very broad distinctions about existence. An upper ontology would allow a distinction like continuant (things that exist, such as objects) versus occurrent (things that happen, such as processes), and hence, provide a way of being more specific about the fundamental differences between the two classes. By functioning in this way, upper ontologies have been proposed as a tool to conceptually unify ontologies that cover a number of different, more specific domains.

Examples of prominent upper ontologies include: The Basic Formal Ontology (BFO) [45], the General Formal Ontology (GFO) [46], the Suggested Upper Merged Ontology (SUMO) [47], and the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [48]. One of the reasons for this diversity and one of the drawbacks of upper ontologies is that each one represents a particular world view derived from a particular branch of philosophical thinking. While the philosophical branch of ontology is a few thousand years old, there are plenty of world views that have not been resolved in that time, and are unlikely to be resolved in the near future.

One of the major claims of upper ontologies is that their use leads to better *ontological modeling*; that is, the knowledge in the ontology is more consistently represented with respect to the distinctions that characterize entities. While understanding how to make fundamental distinctions can be beneficial, there is no way to measure the ontological consistency of the conceptual modeling in a particular ontology, and so the advantage is unproven. The distinctions in upper ontologies are difficult things to master, and there are a lot of notions that are unfamiliar to a domain expert attempting to build an ontology. A biomedical researcher in the process of making useful representations of his or her domain may not need to spend research time learning how to make more accurate high-level distinctions when lower-level distinctions may suffice.

1.6.2 Domain Ontologies

Domain ontologies contain subject matter from a particular domain of interest, for example, biology, physics, or astronomy. Most domain ontologies have a finer granularity than these examples because of the sheer scope of these domains. In biology, for example, we find molecular function, biological process, cellular component (GO), cells [49], biological sequences (Sequence Ontology [50]), and anatomies of various species [51]. When building an ontology to represent the knowledge in a specific domain, it is inevitable that, at the top, there will be some of the most general concepts. For example, a biological ontology may contain the classes organism

and reaction at the top of the hierarchy. In the domain, there are no more general concepts that could be used to structure these classes. For this reason, many domain ontologies are aligned with an upper ontology, so that more fundamental distinctions can be made, for which general classes from the domain are placed underneath the appropriate upper-level class. For example, organism might be mapped to some kind of upper-level class, such as continuant, and glucose metabolism, in contrast, might be mapped to occurrant (a process).

1.6.3 Formal Ontologies

Formality is a much over-used term. It has two meanings in the ontology world. A formal ontology, on the one hand, is one that consistently makes stylistic ontological distinctions based on a philosophical world view, usually with respect to a particular upper-level ontology. On the other hand, formal means to encode meaning with logic-based semantic strictness in the underlying representation in which the ontology is captured [23], thereby allowing computational inferences to be made through the use of automated reasoning. In this case, a formal language is one that allows formal ontologies to be specified, because it has precise semantics. Encoding an ontology in a formal language, however, is not enough to make a formal ontology. For example, it is a common misconception that an ontology encoded in OWL will automatically benefit from computer-based reasoning. It is possible to assert a simple taxonomy in OWL, but without a reasonable usage of a combination of the expressive features provided by OWL, a DL reasoner is unable to make inferences. It is also a common misconception that the use of a DL reasoner will make an ontology better. Description logic can help with the structure, maintenance, and use of the ontology, but it cannot prevent biological nonsense from being asserted (as long as it is logically consistent nonsense).

1.6.4 Informal Ontologies

These are the counterparts of formal ontologies, and informal implies that either no ontological distinctions are made and/or a representation with no precise semantics has been used. Often, the two go together. The lack of ontological formality, or semantic weakness, is not necessarily a bad thing, provided that this is compatible with the intended function of the ontology. Many of the ontologies in this category are what we have already called structured controlled vocabularies.

The goal of these ontologies is to specify a reference set of terms with which the same terms can be used to refer to the same things. The structure in these resources provides a notion of relationships between the terms, most commonly `broaderThan`, `narrowerThan`, and `relatedTo`. Computationally, the relationship amounts to a thing that *has something to do with* another thing. A semantically strict language might state, for example, that each and every instance of this class must have this relationship with at least one instance of this other class and only instances of this class [23]. Sometimes informal ontologies also have more standard relationships, such as *is a* and *part of* relationships [52]. In semantically weak languages, no distinction typically is made between class and instance.

In the context of modern information systems in biomedicine, informal ontologies are frequently applied to the linking, browsing, searching, and mining of information. Controlled vocabularies such as MeSH [53] are semantically weak and make no formal ontological distinctions about the world. They are simply used for indexing and navigating about an information space [37], often a literature database. The actions of searching, browsing, and retrieving information from many different resources are typical of biomedical researchers' practices. Indeed, for the task for which they are intended, the needs of navigation and indexing often contradict strict ontological formality. Thus such so-called ontologies will often be criticized by formal ontologists. This is a mistake, as the intended purpose is different from that of ontologically formal resources.

1.6.5 Reference Ontologies

A distinction can be made between reference and application ontologies [54]. Reference ontologies attempt to be definitive representations of a domain, and are usually developed without any particular application in mind. Reference ontologies will often use an upper-level ontology to make formal (philosophical) ontological distinctions about the domain. They also usually describe one aspect of a domain. The Foundational Model of Anatomy (FMA) [55] is a prime example of a reference ontology for human anatomy. A reference ontology should be well defined, in that each term in the ontology has a definition that should enable instances of that class to be unambiguously identified. Such definitions must at least be in natural language, but can also be made explicit in a semantically strict (computational) representation. As the name suggests, reference ontologies have a primary use as a reference, though they also can be used in an application setting.

1.6.6 Application Ontologies

While reference ontologies are an attempt at a definitive representation of one aspect of a domain, an application domain typically uses portions of several reference ontologies in order to address a particular application scenario. Also, it is often the case that additional information will have to be added to the ontology in order to make the application work. For example, an ontology for describing and analyzing mouse phenotypes might contain a mouse ontology, relevant portions of an ontology describing phenotypes or qualities that can describe phenotypes [56], and an ontology describing assays and other aspects of biomedical investigations [57]. It may also contain aspects of the actual data being represented, the databases in which that data is held, an explanation of what to do with cross-references in the data, and the formats that the data exists in. This particular ontology would most likely contain instance data about individual mice and their measurements, as well as class-level assertions about them. Such a combination of classes and individuals in an ontology is often referred to as a knowledge base.

The Transparent Access to Multiple Bioinformatics Information Sources (TAMBIS) project [9] used an application ontology as a global schema to drive the integration of a series of distributed bioinformatics databases and tools. TAMBIS used an ontology (the TaO) represented in the description logic (DL) GRAIL [58].

The use of a DL allowed automated reasoning to be used, both to help manage the construction of the TaO and to facilitate its use within the TAMBIS application. With a DL and an associated reasoner, axioms within the ontology could be combined to create new descriptions of classes of instances. These descriptions were constructed according to the constraints within the ontology, then classified within the ontology by the reasoner. A class describes a set of instances, and by describing a set of bioinformatics instances in the ontology, a question is being asked. The resources, both tools and databases, underlying TAMBIS were mapped to the TaO, and the conceptual query generated in the TAMBIS user interface was translated to a query against those underlying resources. The larger version of the TaO covered proteins and nucleic acids and their regions, structure, function, and processes. It also included cellular components, species, and publications. A smaller version of the TaO, covering only the protein aspect of the larger ontology, was used in the functioning version of TAMBIS.

1.6.7 Bio-Ontologies

Any one bio-ontology can fall into one or more of the above categories, except the more generic, upper-level ontology. Data integration is a perennial problem in bioinformatics [59]. Bio-ontologies provide descriptions of biological things, and so, when the biological entities referred to in the data are mapped to ontologies that describe the features of those entities, their potential role in data integration becomes obvious. Indeed, the majority of bio-ontologies are used at some level to describe biological data. This is the principle success of the GO, but the use of ontologies as drivers for integration at either the level of schema or the level of the values in the schema are long-standing within bioinformatics and computer science [10]. TAMBIS and EcoCyc (mentioned in Section 1.6.6) were early examples. Once data is described, it can be queried and analyzed in terms of its biological meaning, providing new aspects for looking into the data. As biology is often portrayed as a descriptive science, the role of ontologies in bioinformatics will undoubtedly continue.

1.7 Conclusion

The development and use of bio-ontologies has become an increasingly prominent activity over the past decade, but their main use within bioinformatics so far has been as controlled vocabularies. Terms from these ontologies are used to describe data across many resources, thereby allowing querying and analysis across those resources. Ontologies that harness the power of AI research have been used to start building more intelligent systems that can process data with encoded knowledge and start to support the data-mining process in new ways.

Today, the term *ontology*, itself, includes many forms of structured knowledge that are suitable for addressing different challenges, with elements for human interpretation and for computational inferencing. There remains much disagreement across the community as to exactly what counts as an ontology, and there is an even wider spectrum of opinion about what constitutes a good ontology. Much of this disagreement arises from the diversity of the ontologies and their applications

outlined in this chapter. Tension also arises from the computer-science use of the word and how it differs from the philosophical use.

Whether they are directly describing entities in reality or the information about entities, ontologies are resources that contain computationally accessible, structured knowledge. Such knowledge can be accessed and applied in many research scenarios, such as the data-mining applications described in this book. In essence, in order to successfully mine data, it is necessary to know what the data represents; that is the basic role of ontology within the life sciences. The descriptions that these ontologies provide need to be consistent across the many available data sources and ideally need to be helpful to both humans and computers. With the vast quantity of data now being generated and mined within biology, the need for ontologies has never been greater.

References

[1] Bodenreider, O., and R. Stevens, "Bio-Ontologies: Current Trends and Future Directions," *Brief Bioinform*, Vol. 7, No. 3, 2006, pp. 256–274.

[2] *Oxford English Dictionary: The Definitive Record of the English Language*, http://www.oed.com/, last accessed December 4, 2008.

[3] Guarino, N., "Formal Ontology in Information Systems," *Proc. of FOIS '98*, Trento, Italy, June 6–8, 1998, pp. 3–15.

[4] Musen, M., "Modeling for Decision Support," *Handbook of Medical Informatics*, J. v. Bemmel and M. A. Musen, (eds.), 1997.

[5] Gruber, T. R., "The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases," *Proceedings of KR 1991: Principles of Knowledge Representation and Reasoning*, J. F. Allen, R. Fikes, and E. Sandewall, (eds.), 1991, San Mateo, California: Morgan Kaufmann, pp. 601–602.

[6] Gruber, T. R., *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*, Palo Alto, CA: Knowledge Systems Laboratory, Stanford University, 1993.

[7] Karp, P. D., and M. Riley, "Representations of Metabolic Knowledge," *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, Vol. 1, 1993, pp. 207–215.

[8] Keseler, I. M., et al., "EcoCyc: A Comprehensive Database Resource for Escherichia Coli," *Nucl. Acids Res.*, Vol. 33, Supplement 1, 2005, pp. D334–D337.

[9] Goble, C. A., et al., "Transparent Access to Multiple Bioinformatics Information Sources," *IBM Systems J. Special Issue on Deep Computing for the Life Sciences*, Vol 40, No. 2, 2001, pp. 532–552.

[10] Karp, P., "A Strategy for Database Interoperation", *J. of Computational Biology*, Vol. 2, No. 4, 1995, pp. 573–586.

[11] Schulze-Kremer, S., "Integrating and Exploiting Large-Scale, Heterogeneous, and Autonomous Databases with an Ontology for Molecular Biology," in *Molecular Bioinformatics, Sequence Analysis—The Human Genome Project*, R. Hofestod a. H. Lim, (ed.), Aachen, Germany: Shaker Verlag, 1997, pp. 43–56.

[12] Ashburner, M., et al., "Gene Ontology: Tool for the Unification of Biology," *Nat Genet*, Vol. 25, 2000, pp. 25–29.

[13] Smith, B., et al., "The OBO Foundry: Coordinated Evolution of Ontologies to Support Biomedical Data Integration," *Nat Biotech*, Vol. 25, No. 11, 2007, pp. 1251–1255.

[14] Bodenreider, O., and A. Burgun, "Biomedical Ontologies," *Medical Informatics: Knowledge Management and Data Mining in Biomedicine (Operations Research/Computer Science Interfaces)*, H. Chen, et al., (eds.), New York: Springer-Verlag, 2005.

[15] Bodenreider, O., "Comparing SNOMED CT and the NCI Thesaurus Through Semantic Web Technologies in Representing and Sharing Knowledge Using SNOMED," *Proc. of the 3rd Int. Conf. on Knowledge Representation in Medicine KR-MED 2008*, *CEUR Workshop Proceedings*, Phoenix, AZ, May 21–June 2, 2008.

[16] Corcho, O., M. Fernandez-Lopez, and A. Gomez-Perez, "Methodologies, Tools, and Languages for Building Ontologies. Where is Their Meeting Point?," *Data and Knowledge Engineering*, Vol. 46, No. 1, 2002, pp. 41–64.

[17] Berners-Lee, T., *Weaving the Web,* London: Orion Books, 1999, p. 244.

[18] Berners-Lee, T., J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, Vol. 284, No. 5, 2001, pp. 34–43.

[19] Baader, F., et al., (eds.), *The Description Logic Handbook*, Cambridge, UK: Cambridge University Press, 2003, p. 555.

[20] Horrocks, I., P. Patel-Schneider, and F.v. Harmelen, "From SHIQ and RDF to OWL: The Making of a Web Ontology Language," *J. of Web Semantics*, Vol. 1, No. 1, 2003, pp. 7–26.

[21] K. Wolstencroft, et al., "Protein Classification Using Ontology Classification," *Intellligent Systems for Molecular Biology (ISMB),* Fort a Leza, Brazil, August 6–10, 2006.

[22] Ringland, G. A., and D. A. Duce, *Approaches to Knowledge Representation: An Introduction*, Knowledge-Based and Expert Systems Series, New York: John Wiley, 1998, p. 260.

[23] Aranguren, M., et al., "Understanding and Using the Meaning of Statements in a Bio-Ontology: Recasting the Gene Ontology in OWL," *BMC Bioinformatics*, Vol. 8, No. 1, 2007, p. 57.

[24] Stevens, R., C. A. Goble, and S. Bechhofer, "Ontology-Based Knowledge Representation for Bioinformatics," *Briefings in Bioinformatics*, Vol. 1, No. 4, 2000, pp. 398–416.

[25] Bada, M., et al., "A Short Study on the Success of the GeneOntology," *J. of Web Semantics*, Vol. 1, 2004, pp. 235–240.

[26] Golbreich, C., et al., "OBO and OWL: Leveraging Semantic Web Technologies for the Life Sciences," in *ISWC 2007*, Boston, MA, October 11–13, 2007, pp. 169–182.

[27] Moreira, D. A., and M. A. Musen, "OBO to OWL: a Protege OWL Tab to Read/Save OBO Ontologies," *Bioinformatics*, Vol. 23, No. 14, 2007, pp. 1868–1870.

[28] Day-Richter, J., et al., "OBO-Edit: An Ontology Editor for Biologists," *Bioinformatics*, Vol. 23, No. 16, 2007, pp. 2198–2200.

[29] Horridge, M., et al., "The Manchester OWL Syntax," *OWL Experiences and Directions 2007 (OWLed)*, Insbruck, Austria, June 6–7, 2006.

[30] Rector, A., et al., "OWL Pizzas: Common Errors and Common Patterns from Practical Experience of Teaching OWL-DL," in *European Knowledge Acquisition Workshop (EKAW-2004)*, Northampton, England, October 6–8, 2004, Berlin: Springer Verlag, pp. 63–81.

[31] Cimino, J. J., and X. Zhu, "The Practical Impact of Ontologies on Biomedical Informatics," *Methods Inf Med*, Vol. 45, Supplement 1, 2006, pp. 124–135.

[32] Barrell, D., et al., "The GOA Database in 2009—An Integrated Gene Ontology Annotation Resource," *Nucl. Acids Res.*, Vol. 37, Supplement 1, 2009, pp. D396–D403.

[33] Ogren, P., et al., "The Compositional Structure of Gene Ontology Terms," *Pac Symp Biocomput*, 2004, p. 214–225.

[34] Mungall, C. J., "Obol: Integrating Language and Meaning in Bio-Ontologies," *Comparative and Functional Genomics*, Vol. 5, Nos. 6–7, 2004, pp. 509–520.

[35] Maere, S., K. Heymans, and M. Kuiper, "BiNGO: a Cytoscape Plugin to Assess Pverrepresentation of Gene Ontology Categories in Biological Networks," *Bioinformatics*, Vol. 21, No. 16, 2005, pp. 3448–3449.

[36] Pavlidis, P., et al., "Using the Gene Ontology for Microarray Data Mining: A Comparison of Methods and Application to Age Effects in Human Prefrontal Cortex," *Neurochemical Research*, Vol. 29, No. 6, 2004, pp. 1213–1222.

[37] Bodenreider, O., "The Unified Medical Language System (UMLS): Integrating Biomedical Terminology," *Nucl. Acids Res*., Vol. 32, Supplement 1, 2004, pp. D267–D270.

[38] http://www.nlm.nih.gov/research/umls/, last accessed February 2, 2009.

[39] McCray, A. T., and S. J. Nelson, "The Representation of Meaning in the UMLS," *Methods Inf. Med*., Vol. 34, Nos., 1–2, 1995, pp. 193–201.

[40] McCray, A. T., "An Upper-Level Ontology for the Biomedical Domain," *Comp Funct Genomics*, Vol. 4, No. 1, 2003, pp. 80–84.

[41] Cohen, A. M., and W. R. Hersh, "A Survey of Current Work in Biomedical Text Mining," *Brief Bioinform*, Vol. 6, No. 1, 2005, pp. 57–71.

[42] Hofmann, O., and D. Schomburg, "Concept-Based Annotation of Enzyme Classes," *Bioinformatics*, Vol. 21, No. 9, 2005, pp. 2059–2066.

[43] Yang, J. O., et al., "An Integrated Database-Pipeline System for Studying Single Nucleotide Polymorphisms and Diseases," *BMC Bioinformatics*, Vol. 9, Supplement 12, 2008, pp. S19.

[44] Marquet, G., et al., "BioMeKe: An Ontology-Based Biomedical Knowledge Extraction System Devoted to Transcriptome Analysis," *Stud Health Technol Inform*, Vol. 95, 2003, pp. 80–85.

[45] Smith, B., "Beyond Concepts: Ontology as Reality Representation," *Formal Ontology and Information Systems 2004*, Toring, Italy, November 4–6, 2004, pp. 73–84.

[46] Herre, H., et al., *General Formal Ontology (GFO): A Foundational Ontology Integrating Objects and Processes. Part I: Basic Principles,* 2008, Research Group Ontologies in Medicine (Onto-Med), University of Leipzig.

[47] Niles, I., and A. Pease. "Towards a Standard Upper Ontology," *2nd Int. Conf. on Formal Ontology in Information Systems (FOIS 2001)*, Ogunquit, ME, October 17–19, 2001.

[48] Gangemi, A., et al., "Sweetening Ontologies with DOLCE," *European Knowledge Aquisition Workshop (EKAW-2002)*, Siguenza, Spain, October 1–4, 2002, Berlin: Springer Verlag.

[49] Bard, J., S. Rhee, and M. Ashburner, "An Ontology for Cell Types," *Genome Biol*, Vol. 6, 2005, p. R21.

[50] Eilbeck, K., et al., "The Sequence Ontology: A Tool for the Unification of Genome Annotations," *Genome Biol*, Vol. 6, No., 5, 2005, p. R44.

[51] Baldock, R., A. Burger, and D. Davidson, (eds.), *Anatomy Ontologies for Bioinformatics, Principles and Practice*, London: Springer, 2008, p. 356.

[52] Smith, B., et al., "Relations in Biomedical Ontologies," *Genome Biology*, Vol. 6, No. 5, 2005, p. R46.

[53] Nelson, S. J., et al., "The MeSH Translation Maintenance System: Structure, Interface Design, and Implementation," in *Proc. of the 11th World Congress on Medical Informatics*, San Francisco, September 7–11, 2004, pp. 67–69.

[54] Heijst, V., Shreiber, G. and Wielinga, B. , "Using Explicit Ontologies in KBS," *Int. J. of Human-Computer Studies*, Vol. 46, Nos. 2–3, 1997, pp. 183–292.

[55] Rosse, C., and J. Mejino, "A Reference Ontology for Bioinformatics: The Foundational Model of Anatomy," *J. Biomed Inform*, Vol. 36, 2003, pp. 478–500.

[56] Gkoutos, G.V., et al., "Using Ontologies to Describe Mouse Phenotypes," *Genome Biol*., Vol. 6, r8, 2004.

[57] Patricia L. Whetzel, et al., "Development of FuGO: An Ontology for Functional Genomics Investigations," *OMICS: A Journal of Integrative Biology*, Vol. 10, No. 2, 2006, pp. 199–204.

[58] Baker, P.G., et al., "An Ontology for Bioinformatics Applications," *Bioinformatics*, Vol. 15, No. 6, 1999, pp. 510–520.

[59] Goble, C., and R. Stevens, "State of the Nation in Data Integration for Bioinformatics," *J. Biomedical Informatics*, Vol. 41, No. 5, 2008, pp. 687-693.

[60] Gaunt, J., *Natural and Political Observations Made Upon the Bills of Mortality,* London, 1662.

[61] "ICD-9-CM: International Classification of Diseases, 9th Revision," Clinical Modification, 6th edition, Los Angeles: Practical Management Information Corporation Publisher, 2006.

CHAPTER 2

Ontological Similarity Measures

Valerie Cross

2.1 Introduction

To introduce the topic of this chapter, its title, "Ontological Similarity Measures," first needs an explanation. In this title, the word *measures* is modified by the words *ontological* and *similarity*. Chapter 1 provides an introduction to ontologies. To succinctly summarize, an *ontology* is an explicit specification of a conceptualization [19] that formalizes the concepts pertaining to a domain, the properties of these concepts, and the relationships that can exist between the concepts. As presented in Chapter 1, there are differing levels of complexity, with respect to ontologies, that result in different classifications, ranging from lightweight ontologies to axiomatic ontologies. In deciding to use the word *ontological* in the title of this chapter, the author assumes that the ontology at least has taxonomic relationships between its concepts.

The objective of an *ontological similarity measure* is to determine the similarity between concepts in an ontology. The meaning of the word *similarity* is ambiguous because of its use in many diverse contexts, such as biological, logical, statistical, taxonomic, psychological, semantic, and many more contexts. The context for this chapter is ontological, but the ontological context also falls under the semantic context. An ontological similarity measure is a special kind of semantic similarity measure that uses the structuring relationships between concepts in an ontology to determine a degree of similarity between those concepts. There are other kinds of semantic similarity measures, such as dictionary-based approaches [26, 27] and thesaurus-based approaches [34, 35]. Ontological similarity measures evolved from the early semantic similarity measures based on the use of semantic networks [40].

Determining the semantic similarity between lexical words has a long history in philosophy, psychology, and artificial intelligence. *Syntactics* refers to the characteristics of a sentence, while *semantics* is the study of the meanings of linguistic expressions. A primary motivation for measuring semantic similarity comes from natural-language processing (NLP) applications, such as word sense disambiguation, text summarization and annotation, information extraction and retrieval,

automatic indexing, and lexical selection [7]. Although NLP applications have served as an early motivation for semantic similarity measures, their use has become more widespread because of the need to determine the Semantic Web's degree of interoperability across ontologies, establishing mappings between ontologies, and merge and integrate ontologies from various information systems. More recently, the important role of semantic similarity to bioinformatics research has emerged. Initial experiments [23, 54] and more recent experiments [38, 39] have explored the use of semantic similarity between Gene Ontology (GO) annotations of gene products to determine the similarity between gene products. This semantic similarity assessment between gene products has been compared to the sequence similarity between gene products [1].

Now that the context for *similarity* has been described, the relationship between the three different terms *similarity, distance,* and *relatedness* must be addressed in this context. Sometimes these three terms are used interchangeably in the research literature. These terms, however, are not identical. In determining a semantic relatedness measure, a variety of relationships between concepts in an ontology may be used such as meronymy, synonymy, functionality, associativity, and hyponymy/hypernymy, which is also referred to as subsumption. Semantic similarity is a special case of relatedness that typically uses only the synonymy and the subsumption relationships in the calculation. The meronymy relationships have also been included with the subsumption relationships in determining semantic similarity measures for both WordNet and the Gene Ontology. The relatedness measures, however, may use a combination of the relationships existing between concepts, depending on the context or their importance. For example, the terms car and gasoline are closely related with respect to a functional relationship, but vehicle and car are more similar with respect to the subsumption relationship. Some researchers have made a distinction and have referred to measures between ontological concepts as relatedness measures instead of similarity measures, to emphasize that all relationships between concepts in an ontology may be considered [7]. All semantic similarity measures are, however, semantic relatedness measures.

The term *semantic distance* presents even more difficulty when trying to determine its association with the other two. Much of the research literature supports the view that distance measures the opposite of similarity. Semantic distance, however, could be used with respect to distance between related concepts and distance between similar concepts. In this chapter, for the most part, similarity is the focus, but ontological relatedness and distance measures are discussed when appropriate to the overall presentation.

The remainder of this section first presents an overview of the history behind the development of ontological similarity measures and provides an overview of Tversky's parameterized ratio model of similarity developed using a psychological view of human similarity judgment. Understanding this model is important, since later in the chapter, Tversky's model of similarity is shown to be the basis of many of the ontological similarity measures. The more mathematical presentation of the various categories of these measures is given in Section 2.2 and newer proposed ontological similarity measures are described in Section 2.3.

2.1.1 History

Ontological similarity measures had their beginnings in research to determine the similarity between concepts in a semantic network and their early evaluation in the context of information-retrieval experiments [29, 40]. This initial measure [40] defined semantic distance as the distance between the nodes in the semantic network that correspond to the two words or concepts being compared. The number of edges in the shortest path between the two concepts measures the distance between them. The shorter the distance, the more similar the concepts are.

Rada's approach [40] was based on a hierarchical *is-a* semantic network. Although this edge-counting approach is intuitive and direct, it is not sensitive to the depth of the nodes for which a distance is being calculated. Intuitively, an increase in the depth of the nodes should decrease the distance between the two concepts that are at similar depths in the hierarchy. This weakness was pointed out in an example given in [43]. Figure 2.1, a small example taken from the WordNet ontology [33], clearly illustrates the problem. The distance between *plant* and *animal* is 2 since their common parent is *living thing*. The distance between *zebra* and *horse* is also 2 since their common parent is *equine*. Intuitively, one would judge *zebra* and *horse* as more closely related than *plant* and *animal*. Solely counting links between nodes is not sufficient.

To overcome this limitation of equal-distance edges, numerous researchers proposed various methods to weight each edge. These proposals have resulted in a variety of distance-based ontological measures that are described in detail in Section 2.2.1.

Instead of focusing on distance approaches with adjusted edge weights, other research examined the use of the information content of the concepts in an ontology [41]. The information content for a concept is determined relative to a selected corpus and uses a probabilistic model based on the frequency of occurrence of the concept within the corpus. The similarity between two concepts is then given as a function of the information content of the most specific parent to both concepts. The rationale for this approach is that the similarity between two concepts should be based on the degree to which they share common information. But just as there were critiques of the first proposed distance-based ontological similarity measure, numerous researchers suggested other approaches to using information content for ontological similarity. These approaches integrated not only the shared information

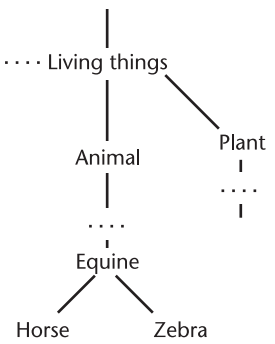


Figure 2.1 Example of WordNet concepts ("..." indicates some concepts are omitted).

content of the two concepts, but also the unshared information content. Details of the information-content approach for ontological similarity are described in Section 2.2.2.

An important part of the research history of ontological similarity measures is the evaluation of these measures. The major evaluation methods of these measures have been mathematical analysis, use in domain-specific applications, and comparison to human judgments on similarity [7]. Mathematical analysis of a measure focuses on investigating its mathematical properties, for example, whether it is a true metric. Very little mathematical-analysis research has been done [31, 56], but more recently, some of the ontological similarity measures have been mathematically compared [13], and this comparison is presented in Section 2.2.3. The use of domain-specific applications for evaluation purposes of ontological similarity measures has primarily been in information-retrieval and NLP applications, but more recently, there has been an explosion in the use of ontological similarity in bioinformatics research. One of the problems in this domain has been how to assess the performance of the various ontological similarity measures. The key approach has been to correlate the ontological similarity measures with other biological methods of similarity assessment. In this situation, the overall similarity between biological objects, such as gene or gene products, is determined based on the ontological similarity of their annotations. The primary correlation has been with sequence similarity [32, 38, 39], but others such as protein-interactions similarity [20], Pfam similarity [11], and gene coexpression [48] have been used. Early in the research on ontological similarity measures, however, the primary way used to compare one measure against another was how well the measure correlates with human judgments of similarity.

Various researchers have carried out experiments using the WordNet ontology to assess the performance of ontological similarity measures. The basis for most of these experiments is a set of 30 noun pairs used in an experiment by Miller and Charles [33]. In the Miller and Charles experiment, the similarity of meaning of each of these 30 noun pairs was rated by 38 human judges. For each pair, the human subjects specified a degree of similarity between the words in a range of 0 to 4, where 0 means no similarity, and 4 means perfect synonymy. These 30 noun pairs were extracted from 65 noun pairs that were used to obtain synonymy judgments by 51 human subjects in a previous experiment that used the same instructions to the human subjects [46]. The 30 pairs were extracted from the 65 pairs in the Rubenstein and Goodenough experiment [46] by selecting 10 pairs with human similarity judgment in the lowest range of 0 to 1, 10 pairs from the medium range of 1 to 3, and 10 pairs from the high range of 3 to 4.

Resnik's investigation of an information-content-based ontological similarity measure [42] was the first to use the word pairs from the Miller and Charles experiment. (It actually used only 28 of the 30 pairs, due to one noun missing from the version of WordNet ontology used in the experiment.) In his experiment, the values of various ontological similarity measures were computed for the 28 word pairs, and coefficients of correlation between the human ratings and the results produced by the various measures were reported. Once this approach to evaluation was documented in the research literature, other researchers followed suit and used these

same word pairs and correlation with the human similarity judgments in order to compare new ontological similarity measures with existing ones [24, 31].

Only a few experiments using the human judgment evaluation method have been performed with a different ontology than WordNet for the calculation of the ontological similarity measures. The primary one is the UMLS ontology [9, 36, 55]. The three source vocabularies from the UMLS Metathesaurus that have been used in these evaluation experiments are MeSH (Medical Subject Heading), SNMI (Systematized Nomenclature of Medicine, also referred to as SNOMED), and the ICD9CM (International Classification of Diseases, Ninth Revision, Clinical Modifications).

Before examining the various traditional approaches to ontological similarity in Section 2.2, the following section examines a foundation model for similarity assessment from the psychology domain because of its importance as the basis for the comparison of ontological similarity measures given in Section 2.2.3.

2.1.2 Tversky's Parameterized Ratio Model of Similarity

In the psychological literature, two main approaches for assessing similarity are content models and distance models. In content models, the characteristics with respect to which objects are similar are conceptualized "as more or less discrete and common elements" [2]. In distance models, these characteristics are conceptualized "as dimensions on which the objects have some degree of proximity." [2]. Many of the proposed set-theoretic measures in the content model category are generalized by Tversky's parameterized ratio model of similarity [53]:

$$S(X,Y) = f(X \cap Y) / \left[f(X \cap Y) + \alpha f(X - Y) + \beta f(Y - X) \right] \tag{2.1}$$

In the above model, X and Y represent sets describing respective objects, x and y. The function f is an additive function on disjoint sets, for example, set cardinality. This measure is normalized so that $0 \leq S(X, Y) \leq 1$. With $\alpha = \beta = 1$, S becomes the Jaccard index [23].

$$S_{\text{jaccard}}(X,Y) = f(X \cap Y) / f(X \cup Y) \tag{2.2}$$

With $\alpha = \beta = 1/2$, S becomes Dice's coefficient of similarity [17]:

$$S_{\text{dice}}(X,Y) = 2f(X \cap Y) / \left[f(X) + f(Y) \right] \tag{2.3}$$

With $\alpha = 1, \beta = 0$, S becomes the degree of inclusion for X: that is, the proportion of X overlapping with Y.

$$S_{\text{inclusion}}(x,y) = f(x \cap y)/f(x) \tag{2.4}$$

Similarly with $\alpha = 0, \beta = 1$, S becomes the degree of inclusion for Y, the proportion of Y overlapping with X. This parameterization is not necessary, however, since

(2.4) can be formulated as $S_{\text{inclusion}}(Y, X)$. It is obvious that the degree of inclusion is not symmetric.

Tversky's research [53] provides direct evidence that human similarity judgments may be directional and therefore, asymmetric. The less salient object is considered more similar to the salient one. Salience is determined by an object's goodness of form or complexity. For example, when asked to rate the similarity between Red China and North Korea, subjects gave a lower similarity for Red China to North Korea, *S*(Red China, North Korea) than that of North Korea to Red China, *S*(North Korea, Red China). When asked to preference the comparison statement, subjects also strongly preferred the comparison order of North Korea to China. The asymmetry is possible, using the above model, whenever $\alpha > \beta$, because the distinctive features of object *X* receive more weight than the distinctive features of *Y*.

Symmetry is not always desired for all similarity assessments. For example, in some cases, a comparison is being made between a subject *u* and a referent *v*: "*u* is similar to *v*." This kind of similarity assessment is extremely relevant to human judgments of similarity. Other psychological research supports asymmetric assessment [6, 22] and justifies the selection of a referent, even when there is not an explicit referent. Different methods have been used to implicitly determine the referent in addition to selecting the object with more salient features [53]. In some cases, there may be a natural reference object or indicator [45]. In others, the degree of an object's prototypicality [22] or informativeness [6] may be used to determine which object serves as the referent.

2.1.3 Aggregation in Similarity Assessment

The Gene Ontology (GO) is playing a significant role in the evaluation of ontological similarity measures. This trend started with the early research found in [32, 54]. The evaluation approach is still the same: that is, to determine how well a measure of gene product similarity that uses ontological similarity correlates with another method of determining similarity, for example, sequence similarity between gene products. This approach, however, introduces another operation beside just ontological similarity measurement between concepts or word pairs, as found in the human-judgment evaluation method. Gene products are annotated with terms or concepts taken from the GO. In ontological approaches that determine the similarity between two gene products, first the ontological similarity between all terms annotated to the first gene product and all terms annotated to the second gene product must be determined. Then, in order to produce an overall semantic assessment of the similarity between the two gene products, these term-pair similarities must be aggregated. The resulting overall semantic assessment of gene-product similarity can then be correlated with the sequence similarity between the gene products. Various approaches to aggregation have been proposed. In [13], an experiment using different aggregation operators with different ontological similarity measures examines the combination effects of the two operations, ontological similarity assessment and aggregation.

To make the following examples and discussion of object similarity concrete, assume that the two objects for which similarity is being assessed are two gene

products, G_X and G_Y, each annotated by a set of terms that exist as concepts in the GO. The sets are $X = \{x_1, x_2, \ldots x_m\}$ for gene G_X and $Y = \{y_1, y_2, \ldots y_n\}$ for gene G_Y. In order to assess how similar G_X and G_Y are, typically two steps are performed:

1. Individual ontological similarity assessment on all the pairs of terms $sim_T(x_i, y_j)$
2. Aggregation of the resulting ontological similarities to produce an overall assessment of the similarity of G_X and G_Y, $sim_A(X, Y)$, where the subscript A specifies the aggregations operator.

For 1, any of the various ontological similarity measures presented in Section 2.2 could be substituted for sim_O. For 2, bioinformatics researchers have applied various aggregation operators on the pairwise similarities, $sim_T(x_i, y_j)$. For example, the simple average [32]

$$sim_{A=PW-ave}(X,Y) = \left[\sum_{i=1,m}\sum_{j=1,n} sim_T(x_i, y_i)\right] \Big/ mn \tag{2.5}$$

or the maximum [59]

$$sim_{A=PW-\max}(X,Y) = \max_{ij}\left[sim_T(x_i, y_j)\right] \tag{2.6}$$

has been used as the aggregation operator to produce an overall similarity measure between G_X and G_Y. Although these approaches for assessing similarity overcome the problem when $X \cap Y = \phi$, they also produce unintuitive results. For example, intuitively, the similarity measure between G_X and G_X, in other words, the genes described by two identical sets, should produce a similarity value of 1. The similarity using the average aggregation operator on the pairwise similarities $sim_{A=PWave}$ does not produce 1. On the other hand, intuitively, the similarity measure between G_X and G_Y described by nonsingleton sets X and Y, having only one element in common, should not produce 1. The similarity using the maximum aggregation operator on the pairwise similarities $sim_{A=PW\text{-}max}$ does produce 1.

A solution to these unintuitive results is the average of the maximum similarity method [3], which combines both the maximum and average aggregation over the pairwise similarities as follows:

$$sim_{A=PW-ave-\max}(X,Y) = \left[\begin{array}{l}\sum_{i=1,m}\max_{j=1,n}\left(sim_T(x_i, y_j)\right) + \\ \sum_{j=1,n}\max_{i=1,m}\left(sim_T(x_i, y_j)\right)\end{array}\right] \Big/ (m+n) \tag{2.7}$$

This pairwise similarity is subscripted as PW-ave-max, because it sums the maximum similarity for each element in X in comparison to the other set Y, and vice versa, and then averages the two.

Other approaches to assessing overall similarity between gene products using GO terms have been proposed. Fuzzy-measure-based ontological similarity [39] is a blend of numerous concepts used in ontological similarity measures. It enlists both the information-content (IC) measure seen in (2.15) and the lowest common ancestor for ontology concepts, as described in Section 2.2. This similarity measure,

however, is not actually an ontological similarity measure in the traditional sense, since it does not assess similarity between concepts in an ontology. Instead, it determines overall similarity between objects described by ontological concepts. One version of the fuzzy-measure-based ontological similarity, S_{AFMS}, augments the sets X and Y with the lowest or nearest common ancestor of every pair (x_i, y_j). For more details on the use of the fuzzy-measure-based approach to assessing similarity between gene products, see [60].

Other proposals for overall similarity assessment of gene products are Sim_{UI} [18] and Sim_{GIC} [37]. Both of these measures are Jaccard set similarity measures, as given in (2.2). In Sim_{UI}, the f function in (2.2) is set cardinality on the sets $X+$ and $Y+$, which are the set of annotations for the two gene products G_X and G_Y, respectively, extended to include not only the original annotations, but also all the ancestors of those annotations based on the GO structure. Sim_{GIC} simply uses the fuzzy-set Jaccard similarity measure, for which instead of weighting each annotation term by 1, the information content of each GO term becomes its weight in the calculation as:

$$sim_{GIC}(G_X, G_Y) = S_{\text{weighted-jaccard}}(X+, Y+) = \sum_{t \in (X+ \cap Y+)} \text{IC}(t) \Big/ \sum_{t \in (X+ \cup Y+)} \text{IC}(t) \quad (2.8)$$

The Section 2.2 now focuses on traditional ontological similarity measures that have been used for the $sim_T(x_i, y_j)$ component. This component produces the individual pairwise ontological similarities that can then be aggregated to produce an overall similarity assessment for two objects, such as gene products G_X and G_Y.

2.2 Traditional Approaches to Ontological Similarity

Although not identical, the terms *similarity* and *relatedness* are connected, in that similarity is a special case of relatedness. The example in [41] illustrates this relationship using the terms *car*, *gasoline*, and *bicycle*. The terms *car* and *gasoline* appear to be more closely related than the terms *car* and *bicycle,* even though *car* and *bicycle* are more similar. This one example shows one kind of relatedness based on a functional relationship such as "car uses gasoline." There are numerous other kinds of semantic relatedness based on the type of relationship between concepts, such as subsumption (e.g., vehicle-car) and meronymy (e.g., car-wheel). The term *semantic distance* presents even more difficulty when trying to determine its association with the other two. Much of the research literature supports the view that distance measures the opposite of similarity. Semantic distance, however, could be used with respect to distance between related concepts and distance between similar concepts. In this chapter, semantic distance signifies the opposite of both semantic similarity and semantic relatedness. The context should provide the basis for the correct interpretation.

2.2.1 Path-Based Measures

Early research focused on using word ontologies to improve information retrieval. One of the most natural approaches [40] to determine semantic similarity in an

ontology is to use its graphical representation and measure the distance between the nodes corresponding to the words or concepts being compared. The number of edges in the shortest path between the two concepts measures the distance between them. The shorter the distance, the more similar the concepts are semantically.

One of the major and intuitively obvious arguments against using the edge-count distance in measuring conceptual distance is the underlying assumption that edges or links between concepts represent uniform distances [43]. In most taxonomic ontologies, concepts that are higher in the hierarchy are more general than those that are lower in the hierarchy. An edge count of 1 between 2 general concepts naturally implies a larger distance than that between 2 more specific concepts. For example, the distance between *plant* and *animal* is 2 in WordNet, since their common parent, is *living thing*. The distance between *zebra* and *horse* is also 2, since their common parent is *equine*. Intuitively, one would judge *zebra* and *horse* to be more closely related than plant and animal. Solely counting links between nodes is not sufficient.

To overcome the limitation of simple edge counting, the edges were weighted to reflect the difference in edge distances. Earlier approaches [25, 29], hand-weighted each edge. Since this approach is not practical for very large ontologies, others proposed methods of automatically weighting each link [43]. As a result of the earlier criticism of simple edge counting, the automatic process was designed to use several pieces of information about the edge in determining its weight: the depth, the density of edges at that depth, and the strength of connotation between parent and child nodes. The weights were reduced as one goes farther down the network, since conceptual distance shrinks. The weight also was reduced in a dense part of the network, since edges in a dense part were considered to represent smaller conceptual distances. Several measures that improve on the original edge-count approach were presented.

One of the simplest adjustments made [28] was to scale the minimum edge count distance between two concepts, $c1$ and $c2$, by the maximum depth D of a taxonomic hierarchy; in other words, this approach uses only hyponymy, or *is-a* type links between concepts.

$$sim_{LC}(c1,c2) = \max\left[-\log\left(\min_{c1,c2}\left[\text{len}(c1,c2)\right]/2D\right)\right] \tag{2.9}$$

Another proposal for conceptual similarity between a pair of concepts, $c1$ and $c2$, [58] scaled the similarity based on the depth of the least common superconcept, $c3$, between $c1$ and $c2$. Although the word *least* was used in the description in [58], the interpretation has been that *least* means *lowest in the ontology*.

$$sim_{WP}(c1,c2) = 2N3/(N1+N2+2N3) \tag{2.10}$$

where $N1$ is the length (in number of nodes) of the path from $c1$ to $c3$, $N2$ is the length of the path from $c2$ to $c3$, and $N3$ is the length of the path from $c3$ to the root of the hierarchy, or, in other words, the global depth in the hierarchy. Conceptual similarity can be converted to conceptual distance as

$$dist_{WP}(c1,c2) = 1 - sim_{WP}(c1,c2) = (N1 + N2)/(N1 + N2 + 2N3) \tag{2.11}$$

In this equation, it is easy to see how an increase of depth decreases the distance between the two concepts.

Another proposal [21] incorporated all relation types in WordNet and, thus, is considered a measure of semantic relatedness. It also incorporated the direction of the link between the two nodes. The directions of links on the same path may vary among horizontal (antonymy), upward (hyponymy and meronymy), and downward (hypernymy and holonymy). Two concepts are semantically related if they are connected by a path that is not longer than an arbitrary fixed constant *C* and that has a direction that does not change too often (where *d* represents the number of changes in path direction and *k* is another constant.)

$$rel_{HS}(c1,c2) = \mathrm{C} - \text{path length}(c1,c2) - kd \tag{2.12}$$

A more sophisticated modification [51] also employed the different kinds of linking relationships within WordNet. Each edge maps to two inverse relations. Each type of relation *r* has a weight range between its own $\min_r$ and $\max_r$. The actual value in that range for *r* depends on $n_r(X)$, the number of relations of type *r* leaving node *X*. This value, referred to as the *type specific fanout* (TSF) factor, incorporates the dilution of the strength of connotation between a source and target node as a function of the number of like relations that the source node has. This factor reflects that asymmetry might exist between the two nodes so that the strength of connotation in one direction differs from that in the other direction. The weight for the relation *r* between nodes *X* and *Y* is calculated as

$$w(XrY) = \max_r - (\max_r - \min_r)/n_r(X) \tag{2.13}$$

and similarly for the inverse relation *r′*, $w(Yr'X)$. The two weights for an edge are averaged. The average is divided by the depth *d* of the edge, within the overall network, to produce the distance or weight between the concepts *X* and *Y* as

$$w(X,Y) = (w(XrY) + w(Yr'X))/2d \tag{2.14}$$

The relative scaling by this depth is based on the intuition that siblings deep in a network are more closely related than only-siblings higher up. The semantic distance between two arbitrary nodes, *c*1 and *c*2, is then computed as the sum of the distances between the pairs of adjacent nodes along the shortest path connecting *c*1 and *c*2.

2.2.2 Information Content Measures

The foundation for this approach is the insight that conceptual similarity between two concepts, *c*1 and *c*2, may be judged by the degree to which they share information [41]. The more information they share, then the more similar they are. In an *is-a* network, this common information is contained in the most specific concept

that subsumes both of $c1$ and $c2$, the common subsumer, which is also referred to as the lowest common ancestor (LCA), $c3$. For example, Figure 2.2 shows a fragment of the WordNet ontology.

The most specific superclass or the lowest common ancestor for *nickel* and *dime* is *coin*, and for *nickel* and *credit card* is *medium of exchange*. The semantic similarity between *nickel* and *dime* should be determined by the information content of *coin,* and that between *nickel* and *credit card* should be determined by the information content of *medium of exchange*. According to standard information theory [61], the information content of a concept c is based on the probability of encountering an instance of concept c in a certain corpus, $p(c)$. As one moves up the taxonomy, $p(c)$ is monotonically nondecreasing. The probability is based on using the corpus to perform a frequency count of all occurrences of concept c, including occurrences of any of its descendents. The information content (IC) of concept c using the corpus approach then is quantified as

$$\mathrm{IC}_{\mathrm{corpus}}(c) = -\log\left(p(c)\right) \tag{2.15}$$

Another approach to determine IC was proposed in [47] using WordNet as an example. The structure of the ontology itself is used as a statistical resource, with no need for external ones, such as corpuses. The assumption is that the ontology is organized in a meaningful and structured way, so that concepts with many descendants communicate less information than leaf concepts. The more descendants a concept has, the less information it expresses. The IC for a concept c is defined as

$$\begin{aligned} IC_{ont}(c) &= \log\left(\left(desc(c)+1\right)/\max_{ont}\right)/\log\left(1/\max_{ont}\right) = \\ &= 1-\log\left(desc(c)+1/\log\left(\max_{ont}\right)\right) \end{aligned} \tag{2.16}$$

where $desc(c)$ is the number of descendants of concept c, and max_{ont} is the maximum number of concepts in the ontology.

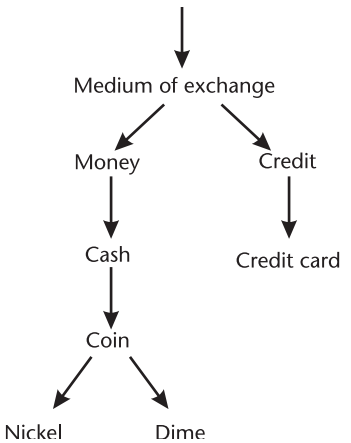


Figure 2.2 A fragment of the WordNet ontology [41].

Resnik proposed the use of information content [41] to determine the semantic similarity between two concepts, $c1$ and $c2$, with the lowest common ancestor of $c3$ as

$$sim_{Res}(c1,c2) = \mathrm{IC}_{\mathrm{corpus}}(c3) \tag{2.17}$$

Note that this is intuitively satisfying since the higher the position of the $c3$ in the taxonomy, the more abstract $c3$ is, therefore, the lower the similarity between $c1$ and $c2$. The lowest common ancestor $c3$, also referred to as the *most specific subsumer*, *most informative common ancestor* [11], or *nearest common ancestor* [39], is the concept from the set of concepts subsuming both $c1$ and $c2$ that has the greatest information content.

While most similarity measures increase with commonality and decrease with difference, sim_{Res} takes only commonality into account. The semantic similarity between two concepts proposed by Lin [31] takes both commonality and difference into account. It uses the shared information content in the lowest-common-ancestor concept and normalizes with sum of the unshared information content of both concepts $c1$ and $c2$, given by

$$sim_{L}(c1,c2) = 2\mathrm{IC}_{\mathrm{corpus}}(c3) \Big/ \left(\mathrm{IC}_{\mathrm{corpus}}(c1) + \mathrm{IC}_{\mathrm{corpus}}(c2)\right) \tag{2.18}$$

Jiang and Conrath [24] began by trying to combine the network distance approach with the information theoretic approach. They envisioned using corpus statistics as a corrective factor to fix the problems with the weighted-edge-counting approaches and developed a general formula for the weight of a link between a child concept c_c and its parent concept c_p in a hierarchy. This formula incorporates the ideas of node depth, local density similar to the TSF, and the link type. These ideas parallel the approach used in [51].

Jiang and Conrath studied the roles of the density and depth components, concluded that they are not major factors in the overall edge weight, and shifted their focus to the link-strength factor. The link-strength factor uses information content, but in the form of conditional probability, or in other words, the probability of encountering an instance of a child concept $c1$, given an instance of a parent concept $c3$. If the probabilities are assigned as in Jiang and Conrath [41], then the distance between concepts $c1$ and $c2$, with concept $c3$ as the most specific concept that subsumes both, is

$$dist_{JC}(c1,c2) = -2\mathrm{IC}_{\mathrm{corpus}}(c3) + \left(\mathrm{IC}_{\mathrm{corpus}}(c1) + \mathrm{IC}_{\mathrm{corpus}}(c2)\right) \tag{2.19}$$

Note that this measure is but a different arithmetic combination of the same term used in the Lin measure given in (2.18). The Lin similarity measure can be converted into a dissimilarity measure by subtracting it from 1 since it is a similarity measure in [0,1] and produces

$$dissim_{L}(c1,c2) = \left[\mathrm{IC}(c1) + \mathrm{IC}(c2) - 2\mathrm{IC}(c3)\right] \Big/ \left[\mathrm{IC}(c1) + \mathrm{IC}(c2)\right] \tag{2.20}$$

The numerator in (2.20) is Jiang-Conrath's distance measure given in (2.19). Jiang-Conrath's distance measure is not normalized. If normalized by [IC($c1$) + IC($c2$)], then it is equivalent to the corresponding dissimilarity measure for Lin's similarity measure.

2.2.3 A Relationship Between Path-Based and Information-Content Measures

Ontological similarity measures have been classified into two primary approaches: path-based and information-content based. In this section, Tversky's parameterized ratio model is used to establish a connection between these approaches. The relationship between two ontological similarity measures found frequently in the research literature is examined. One is taken from the category of distance based within a network structure and the other from those in the information-content category. The connection between these two measures is established through the Tversky's parameterized ratio model, described previously in Section 2.1.2. The following discussion is summarized from [14].

For path-based measures, numerous researchers have emphasized that these are similarity measures based on distances within a taxonomy; however, in Tversky's parameterized ratio model, in particular, the Dice version given in (2.3) can be used to derive both the Wu-Palmer [58] and Lin [31] semantic similarity measures given in (2.10) and (2.18), respectively. Equation (2.21) determines the semantic similarity between $c1$ and $c2$; $c3$ represents the lowest common ancestor of $c1$ and $c2$, and r represents the root, as illustrated by Figure 2.3.

Let X represent the set of *is a* links from r to $c1$ and Y represent the set of *is a* links from r to $c2$. With f simply the cardinality of the sets, the value of $f(X \cap Y)$ represents the cardinality of the intersection between the *is a* links on the path from the root to $c1$ and the *is a* links on the path from the root to $c2$. This value is equivalent to len(r, $c3$) the path length from the root to $c3$, the lowest common ancestor. The assumption is the weights on the links are 1. The value for $f(X)$ is len($c1$, $c3$) + len(r, $c3$), and the value for $f(Y)$ is len($c2$, $c3$) + len(r, $c3$), so that the Dice formula in (2.3) becomes

$$S_{dice/w=1}(c1,c2) = 2\text{len}(r,c3) \Big/ \left(\text{len}(c1,c3) + \text{len}(c2,c3) + 2\text{len}(r,c3)\right) \tag{2.21}$$

which is the Wu and Palmer measure given in (2.10).

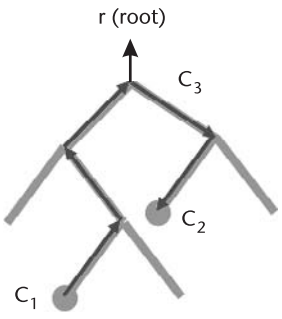


Figure 2.3 Concept hierarchy.

The relationship to Lin's IC semantic similarity measure may be established by modifying weights for the *is a* links from a constant 1 to a weight indicating the strength between the parent and child concepts. The weight of the *is a* links is the difference between the information content of the child node *c* and the information content of the parent node parent(*c*); that is, $w = \text{IC}(c) - \text{IC}(\text{parent}(c))$. This difference indicates how much information is gained by moving from the parent to the child. The len(r, $c3$) then becomes $\Sigma\ (\text{IC}(c) - \text{IC}(\text{parent}(c))) = \text{IC}(c3)$, where c is a node in the path from $c3$ to root r, but it excludes the root, since it does not have a parent. Likewise len($c1$, $c3$) becomes IC($c1$) – IC($c3$) and len($c2$, $c3$) become IC($c2$) – IC($c3$). Substituting these values into (2.21) produces the Lin semantic similarity measure in (2.22).

$$S_{dice/w=IC(c)-\text{IC}(\text{parent}(c))}(c1,c2) = 2\text{IC}(c3)/(\text{IC}(c1)+\text{IC}(c2)) \tag{2.22}$$

Dice's coefficient is the basis for both the Wu-Palmer measure and the Lin measure. The Wu-Palmer measure is easily transformed into the Lin measure by modifying the *is a* link weights from 1 to IC(*c*) – IC(parent(*c*)).

2.3 New Approaches to Ontological Similarity

The traditional approaches to ontological similarity are limited in that they only apply to concepts within the same ontology. Sections 2.3.1 and 2.3.2 describe approaches for assessing similarity between concepts that are in different ontologies. Another limitation is that they only consider the lowest common ancestor. Section 2.3.3 presents an approach to incorporating more than just the lowest common ancestor into the calculation of traditional information content-based ontological similarity measures.

2.3.1 Entity Class Similarity in Ontologies

As shown in Section 2.2.3, Tversky's parameterized ratio model is important to the mathematical comparison of the ontological similarity measures. It also serves as the primary model used in comparisons of entity classes from two different ontologies [44]. This research proposed an ontological similarity measure between entity classes *a* and *b* that uses a matching process over synonym sets, semantic neighborhoods, and distinguishing features. Distinguishing features are further classified into parts, functions, and attributes. The similarity formula used for each matching process is the same and is given as

$$S(a,b) = \left[|A \cap B|\right] \Big/ \left[\begin{array}{l} |A \cap B| + \alpha(a,b)|A - B| \\ +(1-\alpha(a,b)|B - A|) \end{array}\right] \text{ for } 0 \le \alpha \le 1 \tag{2.23}$$

where A and B are description sets that correspond to entity classes a and b (i.e., synonym sets, sets of distinguishing features, or sets of entity classes in the semantic neighborhood). Their only variation from Tversky's model is the method of setting

the α parameter, which weights the relative importance of the noncommon characteristics between the two entities. In their proposal, the α parameter is determined simply from the depth of the entities within their respective ontologies and is given as

$$\alpha\left(a^p, b^q\right) = \begin{cases} \dfrac{\text{depth}\left(a^p\right)}{\text{depth}\left(a^p\right) + \text{depth}\left(b^q\right)} & \text{depth}\left(a^p\right) \leq \text{depth}\left(b^q\right) \\ 1 - \dfrac{\text{depth}\left(a^p\right)}{\text{depth}\left(a^p\right) + \text{depth}\left(b^q\right)} & \text{depth}\left(a^p\right) > \text{depth}\left(b^q\right) \end{cases} \tag{2.24}$$

where entity a belongs to ontology p and entity b to ontology q. This scheme gives priority to the more salient entity, or in other words, the one with the greater depth in its ontology. The various resulting similarities for each of the three categories of synonym sets, sets of distinguishing features, and sets of entity classes in the semantic neighborhood are combined using a weighted aggregation in order to determine the overall similarity between entity classes a and b.

2.3.2 Cross-Ontological Similarity Measures

Traditional ontological similarity measures typically rely only on the hierarchical or *is a* relationships within an ontology. The research in [62] proposes a new method for determining the similarity between two genes or gene products, but it uses two different kinds of relations between terms in the GO. The first category of relations consists of the standard hierarchical *is a* and *part of* relations defined within GO. The second category of relations consists of associative relations created between concepts across the three GO subontologies.

In [63], three nonlexical approaches were used to determine association relationships between GO concepts: (1) a vector space model (VSM); (2) statistical analysis of the co-occurrence of GO terms in annotation databases; and (3) association rule mining. The research in [62] takes advantage of associative relations between GO concepts in different subontologies and uses these associative relations along with hierarchical relationships in order to determine ontological similarity between concepts in the different subontologies, that is, the cross-ontological similarity measure. In their research, the VSM, also referred to as the *cosine method*, is used to determine the strength of association between GO concepts existing in different GO subontologies.

The basic idea of the cross-ontological analysis consists of combining each associative relation across the GO subontologies with a hierarchical relation within a single subontology. Figure 2.4 illustrates the motivation for the cross-ontological similarity measure. Let two ontologies, $O1$ and $O2$, contain concepts $c1$ and $c2$, respectively, and the objective is to determine the ontological similarity between $c1$ and $c2$. Traditional methods would return 0 since the concepts are in two different ontologies. With cross-ontological similarity, however, the associative relationships

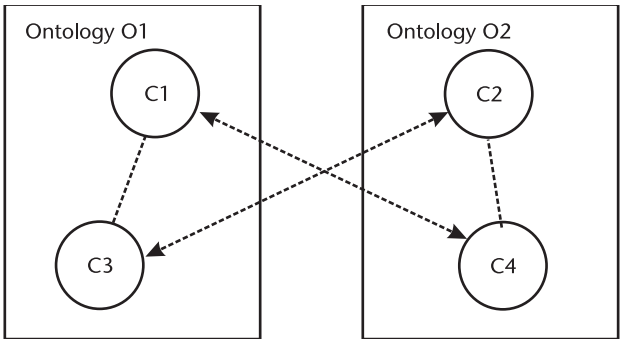


Figure 2.4 Finding similarities between terms across two different ontologies.

between $c1$ in $O1$ and $c4$ in $O2$ and between $c2$ in $O2$ and $c3$ in $O1$ can be used to determine the similarity between $c1$ and $c2$.

The cross-ontological semantic similarity, XOA, between $c1$ and $c2$, can be determined from the traditional or intraontological similarity between concepts $c1$ and $c3$ and between $c2$ and $c4$ and the associative similarities between $c1$ and $c4$ and between $c2$ and $c3$ as

$$XOA(c1,c2) = \max\left\{\max_{c3 \text{ in } O1}\left[\text{sim}(c1,c3)\times\cos(c2,c3)\right], \max_{c4 \text{ in } O2}\left[\text{sim}(c2,c4)\times\cos(c1,c4)\right]\right\} \tag{2.25}$$

In their research, the three information-content measures, Resnik's, Lin's, and Jiang and Conrath's, have been used for the intraontological semantic similarity measure, *sim*, in (2.25). The cos(ci, cj) represents the degree of strength, as determined by the VSM association relationship between concepts ci and cj, which are in different ontologies. Note in (2.25) that cross-ontological similarity is maximizing the trade-off between the best ontological similarity and the best associative similarity within each ontology and then selecting the maximum between the two ontologies.

2.3.3 Exploiting Common Disjunctive Ancestors

In the traditional approaches for ontological similarity, the lowest common ancestor for both concepts, that is, the one with the greatest information content, is critical to both path-based measures and information-content measures. These measures select only the one common ancestor. Others [10, 11] have proposed an ontological similarity measure they categorize as a graph-based similarity measure (GRASM). Instead of simply using the one common ancestor between $c1$ and $c2$ with the greatest information content, the information content of all common ancestors for $c1$ and $c2$ that are not ancestors of any other identified common ancestors are averaged and used in place of the single information content of the lowest common ancestor. The motivation for this approach is that by using just the most informative

common ancestor of $c1$ and $c2$, other shared information content between the two concepts is being ignored. This additional shared information content should be considered in assessing the ontological similarity between $c1$ and $c2$.

The description of this approach [10] first finds disjunctive ancestors of the two concepts for which ontological similarity is being determined. Concepts $a1$ and $a2$ are disjunctive ancestors of c, if there is a path from $a1$ to c not containing $a2$ and a path from $a2$ to c not containing $a1$. Then a common disjunctive ancestor for two concepts, $c1$ and $c2$, is defined as the most informative common ancestor of disjunctive ancestors of $c1$ and $c2$. In other words, $a1$ is a common disjunctive ancestor of $c1$ and $c2$. If, for each ancestor, $a2$ is more informative than $a1$, then $a1$ and $a2$ are disjunctive ancestors of $c1$ or $c2$. This description can be simplified by ignoring the step of finding disjunctive ancestors for each separate concept, $c1$ and $c2$. First, all common ancestors for $c1$ and $c2$ are found. A common disjunctive ancestor for $c1$ and $c2$ is then any common ancestor, ai, of $c1$ and $c2$ that is not the ancestor of any other common ancestor, aj, of $c1$ and $c2$.

To validate this modification to traditional ontological similarity measures, the correlation between ontological similarity assessment of protein families using protein GO annotations and Pfam similarities was investigated. The Pfam database contains the protein families assigned to UniProt proteins [4]. The conclusion of this study was that GraSM provided a consistently higher family similarity correlation across all GO subontologies than traditional ontological similarity measures. To temper this conclusion some, however, the Jiang-Conrath measure was the best performing of the evaluated traditional measures with respect to correlation criteria in these experiments, but its maximum increased correlation, using all common disjunctive ancestors, was only 4% over the three GO subontologies. This small increase in correlation needs to be investigated more in order to determine if significant-enough increases in correlation can be consistently obtained to offset the increased computation needed to find all the common disjunctive ancestors.

2.4 Conclusion

Gene annotation at various levels, from DNA to the cells of an organism, is being accomplished by analyzing and interpreting the data produced by a variety of high-throughput experimental technologies, such as DNA chips and microarrays, protein-protein interaction technologies, proteomics and metabolic profiling for pathway analysis, and many others. The analysis and interpretation of this experimental data often presents a bottleneck to the genome annotation process since human experts are needed to sort through the research literature where most of the experimental results are published. The field of bioinformatics is helping to manage, visualize, integrate, analyze, model, and make predictions from this data.

Methods to automate the annotation process have mainly relied on determining a similarity between a characterized gene or protein and a new one and then predicting its annotations based on its degree of similarity to the known gene or protein. The features of gene products typically used for determining similarity are DNA sequence and expression values. Another more recent approach to assessing similarity between gene products has arisen due to the development of standard

ontologies for the biological and biomedical domains, such as the Open Biomedical Ontologies [50], of which the Gene Ontology is one of the more well known. This approach uses ontological similarity measures between the terms annotating the two gene products to determine their overall similarity. The annotation terms are taken from the controlled vocabulary that is structured in an ontology, such as the Gene Ontology.

As presented in this chapter, various ontological similarity measures, also referred to as semantic similarity measures, were proposed very early on for natural-language-processing applications. Over the last three years, the field of bioinformatics has progressively investigated the use of ontological similarity measures in assessing gene product similarity for a variety of purposes, such as predicting genetic [30] and protein [59] interaction networks, modeling of regulatory pathways [20], validating automatic annotation methods [12], improving the estimation of missing values in microarray data [52], and verifying predictions of protein functions [16]. Ontological similarity measures have established their value in the field of bioinformatics, and continued research in their application, assessing their strengths and weaknesses and developing new approaches to ontological similarity is certain to mature and expand.

References

[1] Altschul, S. F., et al., "Basic Local Alignment Search Tool," *J. Molecular Biology,* Vol. 215, No. 3, 1990, pp. 403–410.

[2] Attneave, F., "Dimensions of Similarity," *American J. of Psychology,* Vol. 63, 1950, pp. 516–556.

[3] Azuaje, F., H. Wang, and O. Bodenreider, "Ontology-Driven Similarity Approaches to Supporting Gene Functional Assessment," *Proc. ISMB 2005 SIG Meeting on Bio-Ontologies,* Detroit, MI, June 21, 2005, pp. 9–10.

[4] Bateman, A., et al., "The Pfam Protein families Database," *Nucleic Acids Research,* Vol. 32, 2004, pp. D138–D141.

[5] Bhattacharyya, P., and N. Unny, "Word Sense Disambiguation and Text Similarity Measurement Using WordNet," *Real World Semantic Web Applications,* V. Kashyap and L. Shklar (eds.), Amsterdam: IOS Press, 2002.

[6] Bowdle F., and D. Gentner, "Informativity and Asymmetric in Comparison," *Cognitive Psychology,* Vol. 34, 1997, pp. 244–286.

[7] Budanitsky, A., "Lexical Semantic Relatedness and Its Application in Natural Language Processing," Computer Systems Research Group, Technical Report, CSRG-390, University of Toronto, 1999.

[8] Budanitsky, A., and G. Hirst, "Evaluating WordNet-Based Measures of Semantic Distance," *Computational Linguistics,* Vol. 32, No. 1, 2006, pp. 13–47.

[9] Caviedes, J., and J. Cimino, "Towards the Development of a Conceptual Distance Metric for the UMLS," *J. of Biomedical Informatics,* Vol. 37, 2004, pp. 77–85.

[10] Couto, F., M. Silva, and P. Coutinho, "Measuring Semantic Similarity Between Gene Ontology Terms," *Data & Knowledge Engineering,* Vol. 61, 2007, pp. 137–152.

[11] Couto, F., M. Silva, and P. Coutinho "Semantic Similarity over the Gene Ontology: Family Correlation and Selecting Disjunctive Ancestors," *Proc. of the ACM Int. Conf. in Information and Knowledge Management,* Bremen, Germany, October 31–November 5, 2005, pp. 43–244.

[12] Couto, F. et al., "GOAnnotator: Linking Protein GO Annotations to Evidence Text," *J. of Biomedical Discovery and Collaboration,* Vol. 1, 2006, pp. 1–19.

[13] Cross, V., "Tversky's Parameterized Similarity Ratio Model: A Basis for Semantic Relatedness," *Proc. of the 2006 Conf. of North American Fuzzy Information Processing Society (NAFIPS),* Montreal, Canada, June 3–6, 2006.

[14] Cross V., and Y. Wang, "Semantic Relatedness Measures in Ontologies Using Information Content and Fuzzy Set Theory," *Proc. 14th IEEE Inter. Conf. on Fuzzy Systems,* Reno, Nevada, May 22–25, 2005, pp. 114–119.

[15] Dagan, I., L. Lee, and F. Pereira, "Similarity-Based Methods for Word Sense Disambiguation," *Proc. of the 35th Annual Meeting of the Assoc. for Computational Linguistics and 8th Conf. of the European Chapter of the Assoc. for Computational Linguistics,* Madrid Spain, July 7–12, 1997, pp. 56–63.

[16] Duan, Z. H., et al., "The Relationship Between Protein Sequences and Their Gene Ontology Functions," *Proc. of the 1st Int. Multi-Symposiums on Computer and Computational Sciences—Volume 1 (IMSCCS'06*), Washington, D.C.: IEEE Computer Society, 2006, pp. 76–83.

[17] Eisler, H., and G. Ekman, "A Mechanism of Subjective Similarity," *Acta Psychologica*, Vol. 15, 1959, pp. 1–10.

[18] Gentleman R., "Visualizing and Distances Using GO," 2005, http://www.bioconductor.org/repository/devel/vignette/GOvis.pdf, last accessed November 15, 2008.

[19] Gruber, T. R., "A Translation Approach to Portable Ontologies," *Knowledge Acquisition*, Vol. 5, 1993, pp. 199–220.

[20] Guo, X., et al., "Assessing Semantic Similarity Measures for the Characterization of Human Regulatory Pathways," *Bioinformatics,* Vol. 22, No. 8, 2006, pp. 967–973.

[21] Hirst, G., and D. St-Onge, "Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms," *WordNet: An Electronic Lexical Database*, C. Fellbaum (ed), Cambridge, MA: The MIT Press, 1998, pp. 305–332.

[22] Holman, F., "Monotonic Models for Asymmetric Proximities," *Journal of Mathematical Psychology,* Vol. 20, 1979, pp. 1–15.

[23] Jaccard, P., "Étude Comparative de la Distribution Florale dans une Portion des Alpes et des Jura," *Bulletin del la Société Vaudoise des Sciences Naturelles,* Vol. 37, pp. 547–579.

[24] Jiang, J., and D. Conrath, "Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy," *Proc. Int. Conf. Research on Computational Linguistics (ROCLING X)*, Taiwan, 1997, pp. 1–15.

[25] Kim, Y., and Kim, J., "A Model of Knowledge Based Information Retrieval with Hierarchical Concept Graph," *J. of Documentation,* Vol. 46, 1990, pp. 113–116.

[26] Kozima, H., and Furugori, T., "Similarity Between Words Computed by Spreading Activation on an English Dictionary," *Proc.6th Conf. of the European Chapter of the Assoc. for Computational Linguistics (EACL-93),* Utrecht, Netherlands, April 21–23, 1993, pp. 232–239.

[27] Kozima, H., and Ito, A., "Context-Sensitive [Measurement of] Word Distance by Adaptive Scaling of a Semantic Space," *Recent Advances in Natural Language Processing: Selected Papers from RANLP'95, Vol. 136 of Amsterdam Studies in the Theory and History of Linguistic Science: Current Issues in Linguistic Theory,* R. Mitkov and N. Nicolov (eds.), Amsterdam: John Benjamins Publishing, 1997, pp. 111–124.

[28] Leacock, C., and Chodorow, M., "Combining Local Context and WordNet Similarity for Word Sense Identification," *WordNet: An Electronic Lexical Database*, Cambridge, MA: The MIT Press, 1998, pages 265–283.

[29] Lee, J. H., Kim, M. H., and Lee, Y. J., "Information Retrieval Based on Conceptual Distance in IS-A Hierarchies," *Journal of Documentation*, Vol. 49, 1993, pp. 188–207.

[30] Lee, P.H., and Lee, D., "Modularized Learning of Genetic Interaction Networks from Biological Annotations and mRNA Expression Data," *Bioinformatics,* Vol. 21, No. 11, 2005, pp. 2739–2747.

[31] Lin. D., "An Information-Theoretic Definition of Similarity," *Proc. 15th Int. Conf. on Machine Learning*, Madison, Wisconsin, July 1998, pp. 296–304.

[32] Lord, P., et al., "Investigating Semantic Similarity Measures Across the Gene Ontology: The Relationship Between Sequence and Annotation," *Bioinformatics*, Vol. 19, No. 10, 2003, pp. 1275–1283.

[33] Miller, G., and Charles, W., "Contextual Correlates of Semantic Similarity," *Language and Cognitive Processes*, Vol. 6, No. 1, 1991, pp. 1–28.

[34] Morris, J., and Hirst, G., "Lexical Cohesion Computed by Thesaurus Relations as an Indicator of the Structure of Text," *Computational Linguistics*, Vol. 17, No. 1, 1991, pp. 21–48.

[35] Okumura, M., and Honda, T., "Word Sense Disambiguation and Text Segmentation Based on Lexical Cohesion," *Proc. 15th Int. Conf. on Computational Linguistics (COLING-94),* Vol. 2, Kyoto, Japan, August 1994, pp. 755–761.

[36] Pedersen, T., et al., "Measures of Semantic Similarity and Relatedness in the Biomedical Domain," *J. of Biomedical Informatics,* Vol. 40, No. 3, 2007, pp. 288–299.

[37] Pesquita, C., et al., "Evaluating GO-based Semantic Similarity Measures," *ISMB/ECCB 2007 SIG Meeting Program Materials, Int. Soc. for Computational Biology,* Vienna, July 21–25, 2007.

[38] Pesquita, C., et al., "Metrics for GO based Protein Semantic Similarity: A Systematic Evaluation," *BMC Bioinformatics Suppl.*, Vol. 5, No. 9, 2008, p. S4.

[39] Popescu, M., J. M. Keller, and J. A. Mitchell, "Fuzzy Measures on the Gene Ontology for Gene Product Similarity," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 3, No. 3, 2006, pp. 263–274.

[40] Rada, R., et al., "Development and Application of a Metric on Semantic Nets," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 19, No. 1, 1989, pp. 17–30.

[41] Resnik, P., "Using Information Content to Evaluate Semantic Similarity," *Proc. 14th Int. Joint Conf. on Artificial Intelligence*, Montreal, Canada, August 1995, pp. 448–453.

[42] Resnik, P., "Semantic Similarity in a Taxonomy: An Information-Based Measure and Its Application to Problems of Ambiguity in Natural Language," *Journal of Artificial Intelligence Research,* Vol. 11, 1999, pp. 95–130.

[43] Richardson, R., Smeaton, A., and Murphy, J., *Using WordNet as a Knowledge Base for Measuring Semantic Similarity Between Word*s. Technical Report, Working paper CA-1294, School of Computer Applications, Dublin City University, Dublin, Ireland, 1994.

[44] Rodríguez, M. A., and Egenhofer, M. J., "Determining Semantic Similarity Among Entity Classes from Different Ontologies," *IEEE Trans. on Knowledge and Data Engineering*, Vol. 15, No. 2, 2003, pp. 442–456.

[45] Rosch, E., "Cognitive Representations of Semantic Categories," *J. of Experimental Psychology*, Vol. 104, 1975, pp. 192–233.

[46] Rubenstein, H., and J. B. Goodenough, "Contextual Correlates of Synonymy," *Computational Linguistics*, Vol. 8, 1965, pp. 627–633.

[47] Seco, N., Veale, T., and Hayes, J., "An Intrinsic Information Content Metric for Semantic Similarity in WordNet," *Proc. European Conf. on Artificial Intelligence*, Valencia, Spain, August 22–27, 2004, pp. 1089–1090.

[48] Sevilla, J., et al., "Correlation Between Gene Expression and GO Semantic Similarity," *IEEE/ACM Trans. on Computational Biology and Bioinformatics,* Vol. 2, No. 4, 2005, pp. 330–338.

[49] Sinha, R., and Mihalcea, R., "Unsupervised Graph-Based Word Sense Disambiguation Using Measures of Word Semantic Similarity," *Proc. of the IEEE Int. Conf. on Semantic Computing (ICSC 2007)*, Irvine, CA, September 2007.

[50] Smith, B. et al., "The OBO Foundry: Coordinated Evolution of Ontologies to Support Biomedical Data Integration," *Nature Biotechnology,* Vol. **25**, 2007, pp. 1251–1255.

[51] Sussna, M., "Word Sense Disambiguation for Free-Text Indexing Using a Massive Semantic Network," *Proc. 2nd Int. Conf. on Information and Knowledge Management,* [http://portal.acm.org/toc.cfm?id=170088&type=proceeding&coll=portal&dl=ACM&CFID=19697427&CFTOKEN=35712577],Washington, D.C., November 1–5, 1993, pp. 67–74.

[52] Tuikkala, J., et al., "Improving Missing Value Estimation in Microarray Data with Gene Ontology," *Bioinformatics,* Vol. 22, No. 5, 2006, pp. 566–572.

[53] Tversky, A., "Features of Similarity," *Psychological Rev.*, Vol. 84, 1977, pp.327–352.

[54] Wang, H., et al., "Gene Expression Correlation and Gene Ontology-Based Similarity: An Assessment of Quantitative Relationships," *Proc. 2004 IEEE Symp. on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB'2004)*, La Jolla, CA, 2004, pp. 25–31.

[55] Wang, Y., "An Empirical Evaluation of Semantic Similarity Measures Using the WordNet and UMLS Ontologies," Master's thesis, Miami University, May 2005.

[56] Wei, M., "An Analysis of Word Relatedness Correlation Measures," Master's thesis, University of Western Ontario, London, Ontario, May 1993.

[57] Wu, J. H., et al., "Prediction of Functional Modules Based on Comparative Genome Analysis and Gene Ontology Application," *Nucleic Acids Res*, Vol. 33, No. 9, 2005, pp. 2822–2837.

[58] Wu, Z., and M. Palmer, "Verb Semantics and Lexical Selection," *Proc. 32nd Annual Meeting of the Assoc. for Computational Linguistics*, Las Cruces, New Mexico, June 1994, pp. 133–138.

[59] Wu, X., et al., "Prediction of Yeast Protein-Protein Interaction Network: Insights from the Gene Ontology and Annotations," *Nucleic Acids Research,* Vol. 34, No. 7, 2006, pp. 2137–2150.

[60] Xu, D., et al., *Applications of Fuzzy Logic in Bioinformatics*, London: Imperial College Press, 2008, p. 225.

[61] Ross, S., *A First Course in Probability*, New York: Macmillan, 1976.

[62] Sanfilippo, A., et al., "Combining Hierarchical and Associative Gene Ontology Relations with Textual Evidence in Estimating Gene and Gene Product Similarity," *IEEE Trans. on Nanobioscience*, Vol. 6, No. 1, 2007, pp. 51–59.

[63] Bodenreider, O., Aubry, M., and Burgun, A., "Non-Lexical Approaches to Identifying Associative Relations in the Gene Ontology," *Proc. Pacific Symp. on Biocomputing*, 2005, pp. 104–115.

CHAPTER 3

Clustering with Ontologies

Mihail Popescu, Timothy Havens, James Keller, and James Bezdek

Clustering, the grouping of objects based on a set of features, is the data-mining area that is, perhaps, one of the most impacted by the proliferation of ontologies. Ontologies have been traditionally used in clustering merely as controlled vocabularies to transform text data in feature vectors in R^N. Recently, ontologies have started to play a transformational role in many clustering algorithms by adding a semantic dimension. As a consequence, many new clustering algorithms have been designed for processing ontological information, and many more have been augmented to include it in their formulation. In this section, we describe several clustering algorithms and cluster-validity measures that take advantage of the semantic similarity (described in Chapter 2) between ontological concepts.

3.1 Introduction

As a knowledge-discovery method, clustering has been employed for many years for exploring datasets for which little information has been available. After the objects in the dataset were grouped, the knowledge was gained either by the guilt-by-association approach (the unknown objects must be similar to the known ones from the same group) or by a reverse-engineering approach (provoked by the question, Why are the objects from this group similar?). Before computers and high-throughput biotechnology devices were available, the datasets were limited in size and complexity. In the postgenomic era, the increase in dataset volume and complexity has fostered the introduction of new clustering methods that increase the selectivity and granularity of the discovered knowledge.

In comparison to engineering datasets, biomedical ones have three distinct characteristics. First, many biomedical objects (genes, patients, etc.) cannot be described as independent objects using vectors in R^N, but, are better described by their relationship to similar objects. The resulting datasets, which we will call in this chapter *relational* (as opposed to *object*), consist of dissimilarity (or distance) matrices. Dissimilarity is the opposite (inverse) of similarity. If there are N objects in our dataset, the dissimilarity matrix, $D = \{d_{ij}\}$, is of size $N \times N$, and d_{ij} represents the dissimilarity between object i and object j. If the data is represented as vectors of real numbers, then d_{ij} corresponds to the distance between vector i and

vector *j*. The best example that illustrates this data type is the computation of the sequence similarity between two DNA fragments. While it is possible to represent each fragment as a four-dimensional vector that contains the count of the {A,C,T,G} nucleotides and use set-based similarity measures (such as Jaccard or Dice, mentioned in Chapter 2), most people use BLAST to find the similarity between the two sequences. Consequently, special relational clustering algorithms are needed to process the resulting dissimilarity matrices. The best-known clustering approach to process relational data is hierarchical clustering. However, many other relational algorithms exist, such as CAST [3], MCL [9], affinity propagation [10], and NERFCM [12]. Among all of the previously mentioned relational clustering algorithms, only NERFCM results in fuzzy cluster memberships; that is, objects can have nonzero membership in multiple clusters. This property is very important in biological problems in which, for example, a gene product may have several functions and may belong to different functional groups. Good reviews of fuzzy relational clustering algorithms can be found in [6, 33].

The second characteristic of biomedical datasets is that, due to their complexity, they often consist of a mixture of categorical and numerical variables, which require specialized dissimilarity measures to compute *D*. Ontologies have started to play an important role in defining new dissimilarity measures that emphasize the semantic relatedness of categorical variables, such as gene function or patient diagnostics. For example, dissimilarity measures based on ontologies, such as Gene Ontology (GO) [33, 36, 37, 30], SNOMED [26, 24], and ICD9 [28], have been defined for computing the dissimilarity between biomedical objects (more examples are provided in Chapter 2). Other dissimilarity measures have been defined by combining numerical dimensions (such as gene expression) with ontological dimensions, as in [15, 21].

The third distinct factor in biomedical datasets is the limited sample size, due to either patient-related issues (such as privacy or disease rarity) or biotechnology issues (such as cost or experimental difficulty). As a result, the datasets tend to have several orders of magnitude more dimensions than the number of samples. To avoid clustering in a sparsely populated high-dimensionality space, these datasets are, most of the time, processed as relational datasets by computing pairwise sample similarities. The typical example for this situation is a microarray dataset that might have only 50–1,000 samples, but 10,000–1,000,000 dimensions (gene fragments). In this case, ontologies (mainly the Gene Ontology) have played an important role in refining the functions of newly found genes by enriching the clustering process. Many examples of clustering applications that have a Gene Ontology component are found on the GO Web site (www.geneontology.org), such as FunCluster [14], GOToolBox [23], or in the literature [2, 15, 21, 34].

The use of ontologies in data mining had a three-prong effect on the study of relational clustering algorithms. First, it stimulated the application of more diverse clustering algorithms to relational data. Several variants of Prim's minimum-spanning tree algorithm, such as CAST [3], VAT [5], and a memetic approach [35], have been developed and applied to GO-based dissimilarity data. In addition, NERFCM [12] has been applied to gene product Gene Ontology dissimilarity matrices [25].

Second, ontologies have stimulated the modification of known nonrelational clustering algorithms to include ontological dissimilarity data (relational data). In

fact, the more general question is whether a nonrelational algorithm (that is, one that requires object data) can be made relational (that is, to accept only dissimilarity data). One approach to handle strictly relational data is NERFCM [12], a relational version of the fuzzy C-means algorithm [4]. A partial transformation of self-organizing maps, OSOM [13], has been reported and applied to gene-product dissimilarity, based on the sets of annotations from the Gene Ontology.

The third effect of ontologies on relational clustering has manifested in the area of cluster validity measures. A *cluster validity measure* tries to answer the question, "How many clusters are in this dataset?". The usual procedure consists in running the clustering algorithm multiple times, varying the requested number of clusters, *C*. Each time, the distribution of the data into the *C* clusters is used to calculate the degree of goodness of the partition (validity measure). Depending on its definition, the value *C* that minimizes or maximizes the validity criterion is chosen as the correct number of clusters. Some cluster-validity measures used in bioinformatics are reviewed in [11]. Other Gene Ontology based cluster-validity techniques can be found in [35] and [7]. In addition, a new cluster-validity measure for fuzzy relational clustering algorithms CCV, was developed in [31].

In this chapter, we describe in more detail three of the algorithms mentioned above, NERFCM, CCV, and OSOM, and provide examples of their application in bioinformatics.

3.2 Relational Fuzzy C-Means (NERFCM)

As with any fuzzy clustering algorithm, NERFCM assigns a set of N objects $O = \{o_1, ..., o_N\}$ to C clusters by computing a fuzzy partition matrix $U = \{u_{ij}\}_{i \in [1,C], j \in [1,N]}$, $U \in M_{fCN}$. Unlike other fuzzy clustering algorithms, NERFCM relies only on the dissimilarities between objects $\{d_{ij}\}_{i,j \in [1,N]}$. The set of fuzzy partition matrices of size $C \times N$, M_{fCN}, can be more exactly described as

$$M_{fCN} = \left\{ U_{C\times N} \left| u_{ij} \in [0,1], \sum_{j=1}^{n} u_{ij} > 0, \forall i \in [1,C]; \sum_{i=1}^{C} u_{ij} = 1, \forall j \in [1,N] \right. \right\} \tag{3.1}$$

The NERFCM algorithm requires that the elements of the dissimilarity matrix (also called the relational matrix) $D_N = \{d_{ij} | i,j \in [1,N]\}$ satisfy the following conditions:

1. $d_{ii} = 0$, for all $i \in [1,N]$;
2. $d_{jk} \geq 0$, for all $j,k \in [1,N]$;
3. $d_{jk} = d_{kj}$, for all $j,k \in [1,N]$.

If the distance matrix D_N were obtained by computing the distance between the objects represented in some feature space $FS \subset R^p$, then D_N would be called *Euclidean*. In general, if D_N were obtained by employing a dissimilarity measure between objects, such as computing the sequence dissimilarity using BLAST [1], it might not be Euclidean. The NERFCM algorithm was especially designed to handle non-Euclidean relational data.

NERFCM is an iterative algorithm that has three main steps. First, an initial guess, U_0, for the fuzzy partition matrix $U = \{u_{ij}\}_{i \in [1,C],\, j \in [1,N]}$, is used to compute C cluster-center vectors, $\boldsymbol{v}_i$, as

$$\boldsymbol{v}_i = \left(u_{i1}^m, u_{i2}^m, \cdots, u_{iN}^m\right) \Big/ \sum_{j=1}^{N} u_{ij}^m, i \in [1, C] \tag{3.2}$$

where $m \in (0,\infty)$ is a parameter (or *fuzzifier*), usually chosen to be 2. Essentially, $\boldsymbol{v}_i$ can be interpreted as a virtual object represented as a mixture of the objects o_j, $j \in [1,N]$. The initial guess, U_0, can be obtained by random initialization, with numbers in [0,1], followed by column normalization.

Second, the dissimilarities d_{ij}, from each object o_j to the ith cluster center, are computed as

$$d_{ij} = \left(D_N \boldsymbol{v}_i\right)_j - 0.5\left(\boldsymbol{v}_t^i D_n \boldsymbol{v}_i\right), i \in [1, C], j \in [1, N] \tag{3.3}$$

where D_N is the dissimilarity matrix between the N objects considered.

Last, an updated fuzzy membership matrix, $U' = \{u'_{ij}\} i \in [1, C], j \in [1, N]$, is computed using

$$u'_{ij} = \begin{cases} \left[\sum_{k=1}^{C} \left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{1}{m-1}}\right] & \text{if } d_{ij} \geq \varepsilon \\ 1 & \text{if } d_{ij} < \varepsilon \end{cases} \tag{3.4}$$

This equation is similar to the related one from FCM [4]. Note, however, that since in (3.4) the dissimilarities are already squared, it does not have the usual 2 in the $1/(m-1)$ exponent. If d_{ij} is smaller than a given ε, $0 < \varepsilon << 1$, u_{ij} is set to 1, and the rest of the memberships in cluster i are set to 0.

If D_N is non-Euclidean, some of the computed distances from (3.3) may be negative, and they could not be used in (3.4). To address this problem, NERFCM uses a β-spread transform [12] that increments, at each iteration, the nondiagonal elements of D_N with a quantity $\Delta\beta$, given by

$$\Delta\beta = \max_{i,j} \left\{2d_{ij} \Big/ \left\|v_i - e_j\right\|^2\right\}, \text{ where } e_j = \left(0, \ldots, \underbrace{1}_{j}, \cdots, 0\right)^t \in R^N \tag{3.5}$$

Accordingly, the distances computed with (3.3) are modified using

$$d_{ij} = d_{ij} + \Delta\beta/2 \left\|v_i - e_j\right\|^2, i \in [1, C], j \in [1, N] \tag{3.6}$$

The distances d_{ij} that are still negative after the above correction are set to 0. The summary of NERFCM is given below.

NERFCM Algorithm: Cluster N samples in C clusters.
Input: D_N = an $N \times N$ dissimilarity matrix
C = the number of clusters

U_0 = an initial guess of the fuzzy membership matrix
MAXIT = maximum number of iterations
DEL = desired fuzzy membership matrix precision
Step 1: Initialize $U = U_0$, it = 0, $\beta = 0$, δ = BIG_NUMBER.
while $\delta <$ DEL and it < MAXIT
Step 2: **for** $i = 1$,C
Compute $\boldsymbol{v}_i$ using (3.2)
end
Step 3: **for** $i = 1$,C and $j = 1$,N
Compute d_{ij} using (3.3)
end
if any $d_{ij} < 0$
compute correction $\Delta\beta$ using (3.5)
modify distances d_{ij} using (3.6)
if some d_{ij} are still < 0, set $d_{ij} = 0$
end
Step 4: recompute fuzzy memberships U' using (3.4)
it = it + 1
$\delta = \|U' - U\|$; $U = U'$;
end
Output: U = a $C \times N$ fuzzy membership matrix.

3.3 Correlation Cluster Validity (CCV)

Correlation cluster validity (CCV) [31] is a validity measure for fuzzy clustering of relational datasets. Assume we want to estimate the number of clusters for a set of N objects $O = \{o_1, ..., o_N\}$, given the dissimilarity matrix $D_N = \{d_{ij}\}_{i,j \in [1,N]}$ between them. For a fixed value C, let U be the final fuzzy partition matrix obtained, say, by running NERFCM on D_N. The main idea of CCV is to define a reconstruction matrix U^* as

$$U^* = 1 - U^t U \Big/ \left(\max\left\{U^t U\right\}\right) \tag{3.7}$$

The assumption used in CCV to find the estimated number of clusters, C, is that the best grouping results in a maximum correlation between U^* and D_N; that is, $C = \arg\max_{i \in [C_1, C_2]}\{corr(U_i^*, D_N)\}$, where U_i^* denotes the reconstruction matrix generated by the $i \times N$ fuzzy membership matrix obtained by grouping the N objects into i clusters. The correlation between the two matrices, (U_i^*, D_N), is computed using the Pearson formula. The summary of the CCV algorithm is given below.

CCV Algorithm: Estimate the number of clusters for N objects.
Input: D_N = an $N \times N$ dissimilarity matrix
m = fuzzifier value
$[C_1, C_2]$ = an interval for searching the number of clusters

for $i \in [C_1, C_2]$ **do**
Step 1: Compute fuzzy memberships for i number of clusters U = NERFCM(D_N, m, i)
Step 2: Compute the reconstruction matrix U_i^* using (3.7)
Step 3: corr(i) = Pearson(U_i^* ,D_N)
end
Step 4: $C = \arg\max_{i \in [C_1, C_2]} \{corr(i)\}$;
Output: C = the estimated number of clusters.

3.4 Ontological SOM (OSOM)

One way that researchers have dealt with conventional high-dimensional datasets is to employ self-organizing maps (SOM), as initially proposed by [18–20]. The SOM allow these types of data to be effectively visualized in two or three dimensions, by combining the goals of both projection and clustering algorithms [16]. In [13], we described a novel extension to the SOM that allows us to use the SOM with ontological data.

The self-organizing map is a two-layer, lateral feedback neural network that topologically maps itself to the training data. The network structure is often set to a two-dimensional rectangular, toroidal, or hexagonal grid of P nodes, where each network node (neuron) $\boldsymbol{a}_i \in R^2$, $i \in [1, P]$, is laterally connected to its neighbors. Assume we want to cluster a set of N objects $O = \{o_1, ..., o_N\}$ represented by M ontology terms. Each object o_i is represented as a binary vector $\boldsymbol{x}_i = (x_{i1}, x_{i2}, ..., x_{iM})$ with $x_{ij} \in \{0,1\}$, where $x_{ij} = 1$, if the object i is annotated with the jth ontology term, and M is the total number of ontology terms available. Each node $\boldsymbol{a}_i$ is connected to a prototype $\boldsymbol{w}_i \in R^M$ from the feature space. The standard SOM network learning algorithm is iterative, and it has two main steps [19]. In the first step, the closest SOM prototype, $\boldsymbol{w}_p$, to a randomly drawn sample from the data, $\boldsymbol{x}_d$, is updated using

$$\boldsymbol{w}_p^{(new)} = \arg\min_i \left\{ \left\| \boldsymbol{x}_d - \boldsymbol{w}_i^{(old)} \right\| \right\} \tag{3.8}$$

where ||·|| is any distance metric, and $\boldsymbol{w}_i^{(old)}$ are the prototype values from the previous iteration. In the second step, the SOM prototypes for nodes in a neighborhood of the node $\boldsymbol{a}_p$ connected to $\boldsymbol{w}_p$ are updated by

$$\boldsymbol{w}_i^{(new)} = \boldsymbol{w}_i^{(old)} + \varepsilon(t) \cdot h_{ip} \cdot \left(\boldsymbol{x}_d - \boldsymbol{w}_i^{(old)} \right) \tag{3.9}$$

where ε is the learning rate. In (3.9) above, h_{ip} is a neighborhood function defined as

$$h_{ip} = \exp\left(\frac{-1 \left\| \boldsymbol{a}_i - \boldsymbol{a}_p \right\|^2}{\sigma^2(t)} \right) \tag{3.10}$$

that is, $\boldsymbol{a}_i$ are SOM nodes in a Gaussian-shaped neighborhood of $\boldsymbol{a}_p$ (e.g., a square or hexagonal grid) with a radius determined by the variance σ^2(t).

This algorithm is repeated until a maximum number of iterations are completed, or there is no change in the position of the nodes $\boldsymbol{a}_j$, $j \in [1,P]$. Typically, the learning rate $\varepsilon(t)$ and the radius of the neighborhood function $\sigma^2(t)$ are reduced during iteration, with the effect that late iterations are only updating network prototypes local to the winning prototype $\boldsymbol{w}_p$.

The ontological self-organizing map, OSOM, proposed in [13] is an adaptation of the standard SOM to ontological data. First, we construct an ontological prototype vector $\boldsymbol{w}_i \in [0,1]^M$, $i \in [1,P]$, for each node in the OSOM grid. Each prototype vector element, w_{ij}, represents the contribution of the ontology term j to the description of the associated node $\boldsymbol{a}_i$.

Second, we replace the distance metric in step 1 of the SOM (3.8) with an ontology-based dissimilarity measure. The measures we may use [13] are vector-matrix multiplication-based operations that are simple extensions of the measures described in Chapter 2. For example, the average dissimilarity is defined as

$$S^{(AVG)}\left(\boldsymbol{w}_i, \boldsymbol{x}_j\right) = \frac{\boldsymbol{w}_i^t D_M \boldsymbol{x}_j}{M\left|\boldsymbol{x}_j\right|} \tag{3.11}$$

where D_M is the term-dissimilarity matrix computed using one of the methods presented in Chapter 2 (e.g., an information-content measure) and $\boldsymbol{x}_j$ is the jth data vector. Note that the presence of the dissimilarity matrix, D_M, in (3.11) provides a mapping among the dimensions of the prototype vectors $\boldsymbol{w}_i$.

Finally, we replace the prototype-vector update equation in step 2 of the SOM with a dissimilarity-based update. In order to create the new update equation, we define two axioms that the equation must satisfy:

Axiom 1. The prototype vectors must move closer to the randomly chosen training data vector $\boldsymbol{x}_d$, $d \in [1, N]$, at each iteration.

Axiom 2. The prototype vectors must also move closer to terms that are similar to the terms in $\boldsymbol{x}_d$, $d \in [1, N]$.

With these axioms in mind, we created the following update equation:

$$\boldsymbol{w}_i^{(new)} = \boldsymbol{w}_i^{(old)} + \varepsilon(t) \cdot h_{ip} \cdot \left(F\left(D_M, \boldsymbol{x}_d\right) - \boldsymbol{w}_i^{(old)}\right) \tag{3.12}$$

where p denotes the closest OSOM prototype to the randomly chosen training vector $\boldsymbol{x}_d$, i denotes the nodes from its neighborhood and $F(D_M, \boldsymbol{x}_d)$ is an update operator. The update operator is computed from the columns of the dissimilarity matrix that correspond to nonzero elements of the training vector $\boldsymbol{x}_d$. These columns of the dissimilarity matrix represent the dissimilarity between the nonzero terms in $\boldsymbol{x}_d$ and all other terms (e.g., $D_{M,ii}$ because a term is perfectly similar to itself). Hence, the update operator $F(D_M, \boldsymbol{x}_d)$computes a row aggregation on the dissimilarity matrix D_M and training vector $\boldsymbol{x}_d$, producing the update step for the OSOM prototypes. The operator F can be defined using a MAX operator as

$$F_k^{(MAX)}(D_M, \boldsymbol{x}_d) = \max_i \{1 - D_{M,ki}\} \tag{3.13}$$

where $i = \{j | j \in [1, M]; x_{dj} = 1\}$, $k \in [1,M]$; that is, we consider not only the terms that annotate $\boldsymbol{x}_d$, but also the degree to which the other M terms are similar to them. In other words, $F^{(MAX)}$ pushes the OSOM prototypes toward the terms present in $\boldsymbol{x}_d$ and, additionally, pushes the prototypes toward all the terms represented in D_M that are similar to any one of the terms in $\boldsymbol{x}_d$. Other aggregation operators can be found in [17].

An outline of the OSOM algorithm is given below.

OSOM Algorithm: Visualize the grouping of N objects based on their ontology description.
Input: $-\{\boldsymbol{x}_i\}$, $i \in [1,N]$, $\boldsymbol{x}_i \in [0,1]^M$ = a set of N vectors that describe objects using M ontology terms.

- D_M = a dissimilarity matrix between the M ontology terms (see Chapter 2)
- P = the number of nodes in a grid
- grid topology
- initial and final learning rate: ε_0, ε_f ; initial and final radius: σ_0, σ_f
- $\{\boldsymbol{w}_i^0\}$ = a random initialization of prototype weight vectors $\boldsymbol{w}_i \in [0,1]^M$
- $t_{\max}$ = a maximum number of iterations
- set $t \leftarrow 0$

while $t < t_{\max}$ **do**

1. Randomly draw a single training data vector, $\boldsymbol{w}_i$.
2. Find closest prototype, $\boldsymbol{w}_p = \arg\min_i S(\boldsymbol{w}_i, \boldsymbol{x}_d)$ using (3.11).
3. Update prototype vectors with (3.12).
4. Decrease the neighborhood size: $\sigma(t) = \sigma_0(\sigma_f/\sigma_0)^{t/t_{\max}}$.
5. Decrease the learning rate $\varepsilon(t) = \varepsilon_0(\varepsilon_f/\varepsilon_0)^{t/t_{\max}}$.
6. $t \leftarrow t+1$

end
Output: a mapping of the N objects defined by the position of the P nodes of the grid.

The parameters, such as the learning rate and maximum iterations, are set according to the specific problem.

3.5 Examples of NERFCM, CCV, and OSOM Applications

3.5.1 Test Dataset

The basis of our illustrative computations is a set of 194 human gene products [30] that were clustered into three protein families using Markov clustering (MCL) [9]. The gene products (and the related information) were retrieved on December 10, 2003 using the ENSEMBL browser (http://www.ensembl.org/). In Table 3.1, we give a summary of the dataset, called GPD_{194}, containing information on these families.

Table 3.1 Summary of the GPD_{194} Dataset

ENSEMBL Family (ENSF)	F_i = *Protein Family*	*No. of Genes*	N_i = *No. of Sequences*
339	Myotubularin	7	21
73	Receptor precursor	7	87
42	Collagen alpha chain	13	86

The GPD_{194} dataset has several noteworthy characteristics: (1) each group has multiple well-characterized genes, many of which are involved in human disorders when mutated, and all of which could be considered very similar in both structure and function; (2) several of the genes, especially the receptor precursor genes, are characterized by multiple isoforms represented by multiple sequences, and thus represent extremely similar gene products; (3) the MCL clustering available through ENSEMBL pulled together these gene groups, allowing us a cluster method by which to compare our results. The myotubularins have protein tyrosine phosphatase enzymatic activity, are involved in dephosphorylation, and are active in muscle tissue. The receptor precursor proteins are integral to the plasma membrane and are involved in the fibroblast growth-factor-signaling pathway influencing cell division and cell differentiation. The collagen alpha-chain genes are involved in producing the alpha chain of type 1 collagen that adds strength and structure to connective tissue found in ligaments, bones, and cartilage.

3.5.2 Clustering of the GPD_{194} Dataset Using NERFCM

The GPD_{194} gene products were annotated at the time of retrieval (December 10, 2003) by a total of 64 GO terms. Each individual gene product was annotated by between 2 and 10 GO terms. The purely relational dissimilarity matrix D_{FSM} (GPD_{194}) between the 194 gene products from GPD_{194} computed using the fuzzy measure dissimilarity [30] on the set of GO annotations is shown in Figure 3.1. We mention that the above similarity could have been computed using any of the GO similarity methods described in Chapter 2.

Figure 3.1 shows that the gene products indexed 1 to 21 seem to be strongly similar to each other and not to the remaining gene products, indicating that they should end up in a single cluster. Similarly, the gene products indexed 22 to 108 and 109 to 194 show dissimilarity patterns that seem to be related to the ones described in Table 3.1. Of course, we have ordered the gene products so that this similarity structure is apparent to the reader. In general, this dissimilarity matrix is presented to a clustering algorithm in a randomized manner. After running NERFCM with $C = 3$ and $m = 2$, the cluster memberships shown in Figure 3.2 seem to tell another story.

The memberships in cluster 1 are the greatest for gene products with indexes from 22 to 108, suggesting that these gene products belong to the same cluster (cluster 1, row 2 in Table 3.1); however, gene products indexed 1 to 21 and 154 to 194 seem to be clustered together (membership in cluster 2 is the greatest for those indexes). Moreover, the third group from Table 3.1 (gene products 109 to 194) is

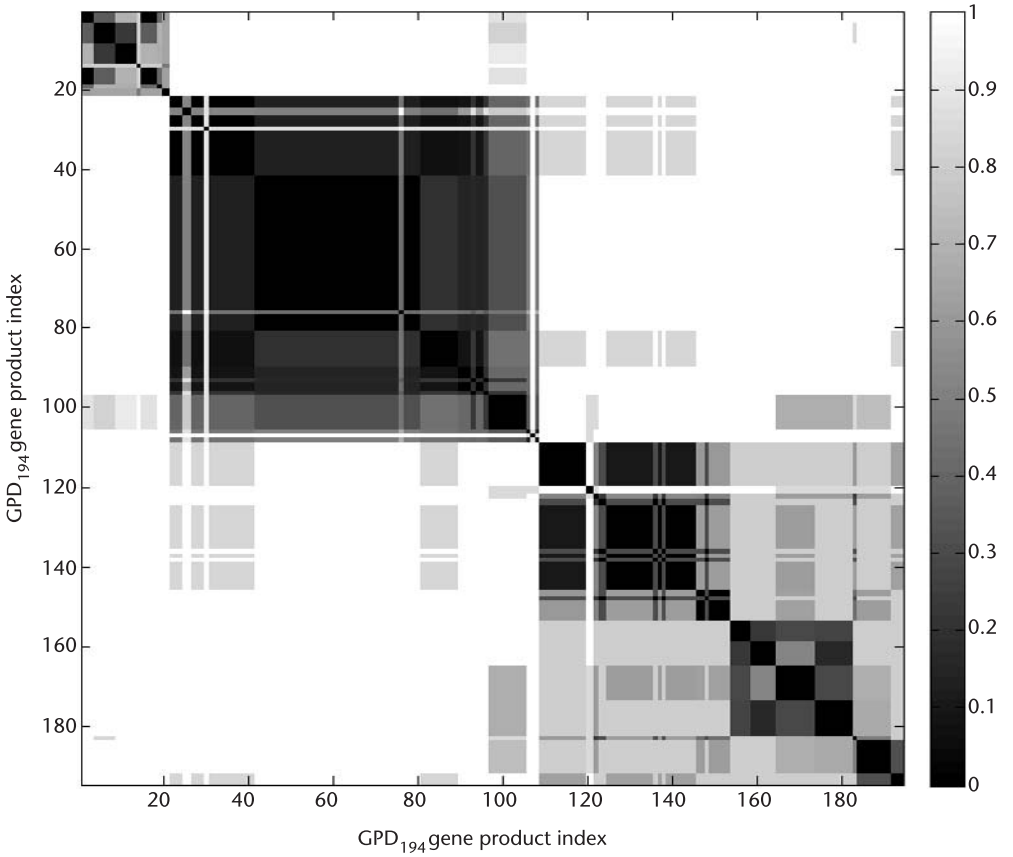


Figure 3.1 The fuzzy measure dissimilarity matrix, $D_{FSM}(\text{GPD}_{194})$, for the GPD_{194} GO annotation dataset.

split in two clusters, cluster 2 (indexes 154 to 194) and 3 (indexes 109 to 153). These observations suggest that GPD_{194} does not contain 3 clusters as claimed by ENSMBL in 2003, which leads us to the question: How many clusters are in the GPD_{194} dataset?

3.5.3 Determining the Number of Clusters of GPD_{194} Dataset Using CCV

To estimate the number of clusters present in the GPD_{194} dataset, we use the CCV algorithm described in Section 3.3. The algorithm consists of repeatedly applying a fuzzy clustering algorithm (NERFCM, in our case) for different numbers of clusters and computing the correlation between each resulting reconstruction matrix (see (3.7)) and the original dissimilarity matrix.

The plot of the correlation value at different numbers of clusters for the GPD_{194} dataset is shown in Figure 3.3. The maximum correlation, about 0.95, is obtained for $C = 5$, suggesting that GPD_{194} contains 5 clusters. These clusters from NERFCM with $C = 5$ and $m = 2$ are shown in Figure 3.4, delimited by a dotted line.

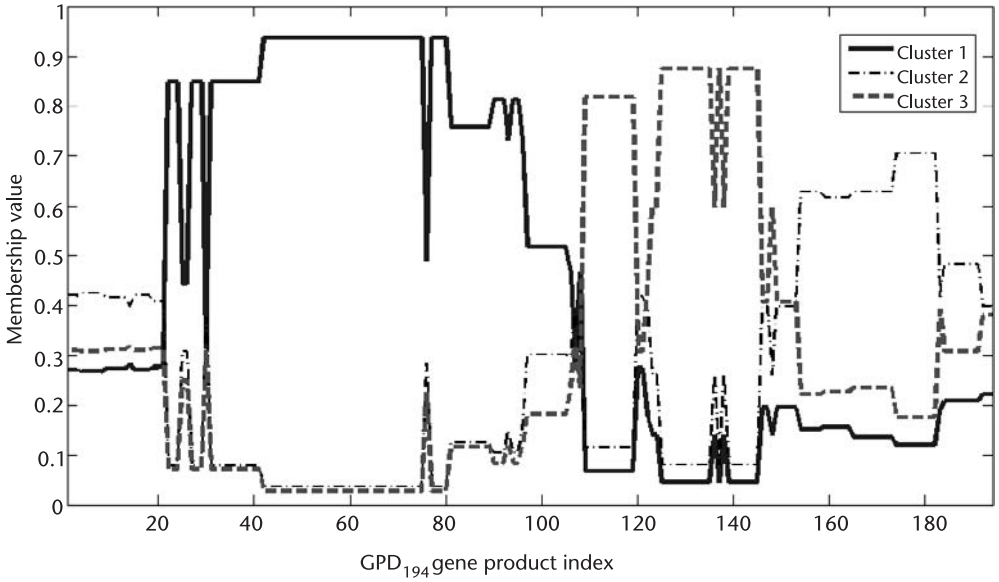


Figure 3.2 The fuzzy memberships for the GPD_{194} dataset, computed using NERFCM clustering algorithm with $C = 3$ and $m = 2$.

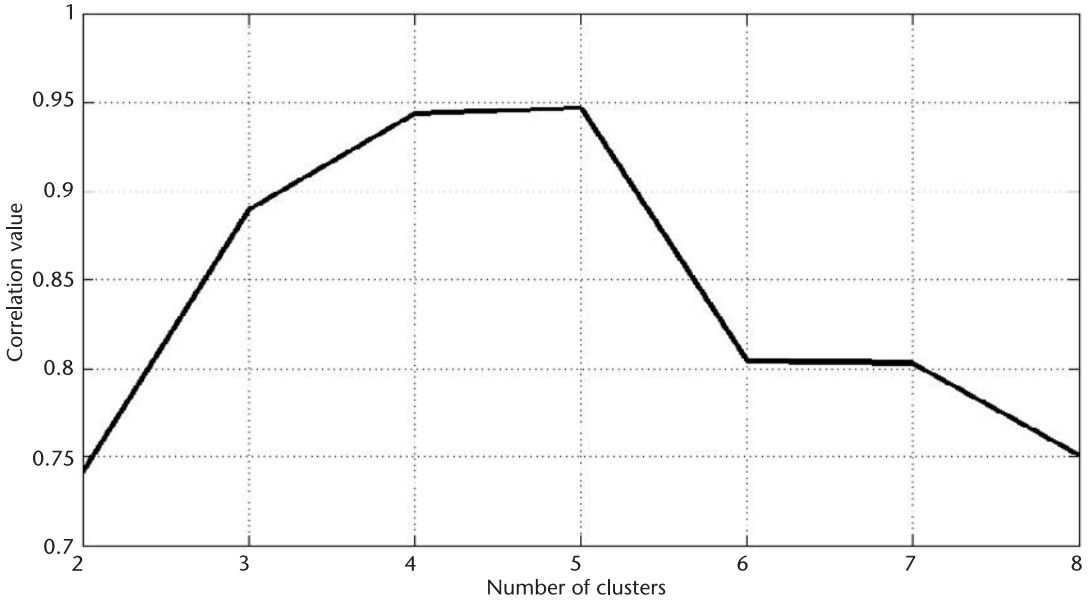


Figure 3.3 The CCV correlation values versus the number of clusters for the GPD_{194} dataset obtained using NERFCM with $m = 2$.

With few exceptions, the indexes of the gene products from Figure 3.4 correspond to those in Figure 3.1 (cluster 1 pulled in a few extra gene products). Here, CCV found that the third cluster in Figure 3.1 (third row in Table 3.1, representing the collagen family) contains 3 subclusters. This finding has been verified by

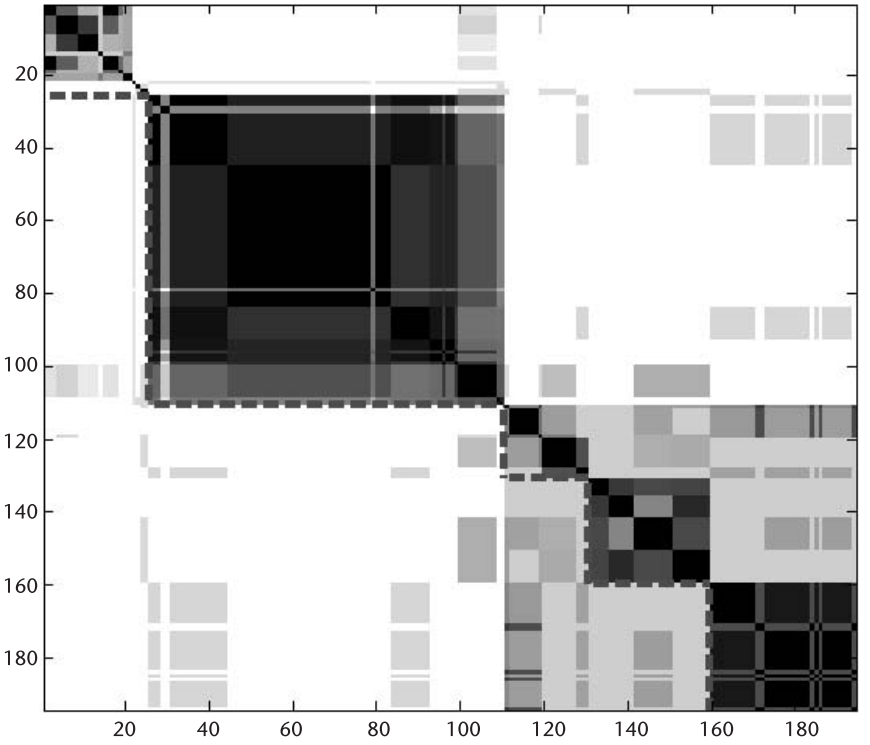


Figure 3.4 The five clusters identified in the GPD_{194} dataset from NERCM.

[22]. From the 9 collagen families mentioned by Myllyharju and Kivirikko [22], we have in our dataset only 3 collagen subfamilies: fibril-forming collagens (FFC): *{COL1A1, COL2A1, COL3A1, COL5A3, COL24A1, COL27A1}*, type IV collagens {COL2A1, COL2A2, COL2A3, COL2A6}, and the fibril-associated collagens with interrupted triple helices (FACIT) *{COL9A1, COL9A2, COL21A1}*, which are exactly the 3 subclusters found by CCV [30].

3.5.4 GPD_{194} Analysis Using OSOM

We apply our ontological self-organizing map (OSOM) to produce cluster visualization and functional summarization of the GPD_{194} dataset.

3.5.4.1 GPD194 Visualization Using OSOM

We applied the OSOM algorithm described in Section 3.4 using a toroidal grid-based network with $P = 400$ neurons (a 20×20 matrix). The learning rates are $\{\varepsilon_0 = 0.5, \varepsilon_f = 0.005\}$, the radii of the lateral influence function in (3.10) are $\{\sigma_0 = 3.0, \sigma_f = 0.1\}$, and the maximum number of iterations is $t_{\max} = 10{,}000$.

The visualization method maps the *gene-product profiles* (the OSOM prototypes) of the OSOM network to the nodes of the two-dimensional toroidal grid (see Figure 3.5).

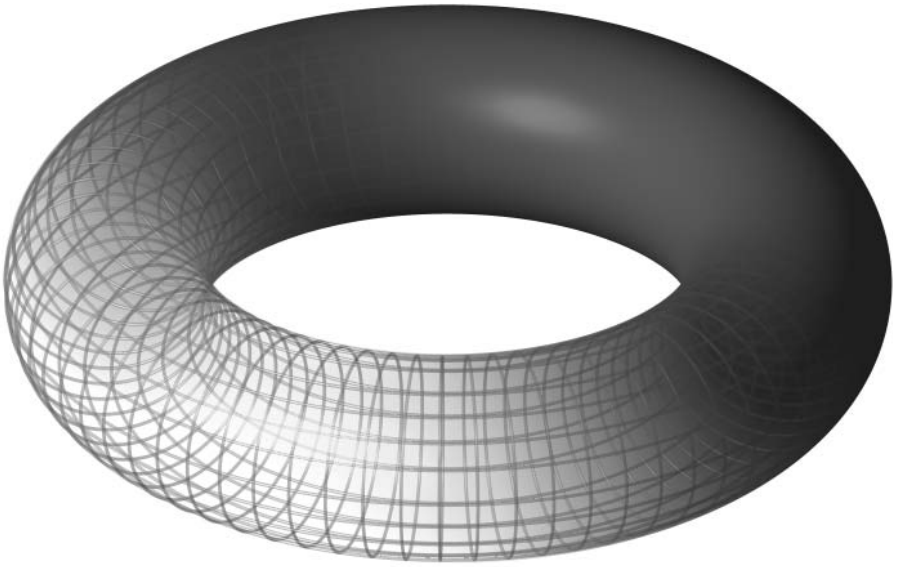


Figure 3.5 The toroidal grid used in the GPD_{194} OSOM representation.

To show the cluster tendency of gene products, the relations between neighboring gene -product profiles on the grid are displayed as gray levels—black representing *no relation* and white representing *highly related*.

The visualization method we propose is composed of two distinct steps. (1) the gene products are mapped to the trained OSOM network by the nearest prototype rule—for each gene product $\boldsymbol{x}$, find the best match prototype $\boldsymbol{w}_p = \arg\min_{i\in[1,P]}\{S(\boldsymbol{w}_i, \boldsymbol{x})\}$. In this fashion, the node p of the network is associated with the gene product $\boldsymbol{x}$. As a result, similar gene products are mapped to groups of similar nodes in the network; (2) the similarity between neighboring OSOM nodes is mapped into a grayscale image—white showing high dissimilarity, black showing very low dissimilarity [16]. Figure 3.6(a) illustrates this mapping using the AVG dissimilarity operator (3.11) and MAX update operator (3.13). The white regions correspond to groups of similar gene product, while the black regions show the boundaries between groups that are dissimilar. Please note that, due to the toroidal topology of the OSOM network, the top and bottom, as well as the sides, wrap around.

The dissimilarity between nodes is then calculated by an average operator

$$S^{(OSOM)}\left(\boldsymbol{w}_i, \boldsymbol{w}_j\right) = \frac{\boldsymbol{w}_i^t D_M \boldsymbol{w}_j}{M^2} \tag{3.14}$$

And this dissimilarity is calculated between each node of the OSOM network in the up-down, left-right, and four diagonal directions. Thus, each prototype node has eight surrounding pixels that correspond to its dissimilarity to neighboring nodes. The grayscale color map is set such that white corresponds to $\max_{\forall i,\forall j}[S^{(OSOM)}(\boldsymbol{w}_i, \boldsymbol{w}_j)]$ and black corresponds to $\min_{\forall i,\forall j}[S^{(OSOM)}(\boldsymbol{w}_i, \boldsymbol{w}_j)]$ for a given network, where $i \in [1,N_H]$, $j \in [1,N_V]$, and N_H, N_V are the horizontal and

vertical dimensions of the grid, respectively (in our case, $N_H = 20$, $N_V = 20$). The color at the node location is interpolated from the eight surrounding pixels.

As a result of this coloring method, regions that are lightly colored represent groups of similar gene products, while darker regions signify outliers or gene products that are dissimilar to the surrounding groups. In addition, the degree of dissimilarity can be seen in the intensity of the regions. For example, in Figure 3.6(a), the light region on the right is a highly similar group, while the more gray regions signify dissimilarity to a lesser degree, and the black regions denote boundaries between dissimilar groups of gene products. In contrast to OSOM, in Figure 3.6(b), we show the same map obtained using the regular SOM, that is, the SOM where no ontological similarity was used.

The three GPD_{194} families can be seen in Figure 3.6(a) as light-colored islands. The *collagen alpha chains* are located in the top-left and bottom-left (recall that the grid is toroidal; hence, these two regions are actually connected). The *myotubularins* are located at the top-right and bottom-right. Lastly, the *receptor precursors*, which are the most tightly grouped gene products (they are mapped to a bright region), are located at the right-middle of the image. We note that the *TEK* gene was mapped into 2 nodes (10, 3) and (19, 10). This was due to the fact that, in this version of GO annotations, the gene product mapped to the node (10, 3) had the wrong annotation. In contrast, each family is broken in 2–4 pieces in the SOM map, as shown in Figure 3.6(b).

3.5.4.2 Functional Summarization of Gene Product Clusters

Functional summarization of the gene-product profiles is achieved by examining the OSOM prototype weight vectors. The ontological content of each OSOM prototype is represented by a vector, as discussed in Section 3.4. Each element of the prototype vector can be viewed as the influence of a specific GO annotation in defining the profile of its associated OSOM node. Thus, high values in a prototype vector signify a high likelihood that the gene products mapped to that location in the OSOM are annotated by that specific term or by a term that is very similar, according to the specified term-based dissimilarity measure. We define the most representative term (MRT) of a gene-product profile as the term that has the highest associated weight in the OSOM prototype vector.

The strength of the OSOM visualization method is that it shows the overall dissimilarity of the genes as seen by the three distinct islands, which represent the three families. However, groups are mapped to different locations due to minor differences in their ontological data. In Table 3.2, we present the MRTs for the entire trained OSOM network, as shown in Figure 3.6(a).

The terms from the Table 3.2 represent a functional summarization of all the gene-product groups present in the GPD_{194} dataset. The dataset has been summarized using the following eight GO terms: protein amino acid dephosphorylation, extracellular matrix structural constituent, kinase activity, receptor activity, protein-tyrosine kinase activity, ATP binding, cell adhesion, and collagen type IV. The gene summarization was performed using only 8 of the 64 GO terms used in the annotation of the GPD_{194} dataset.

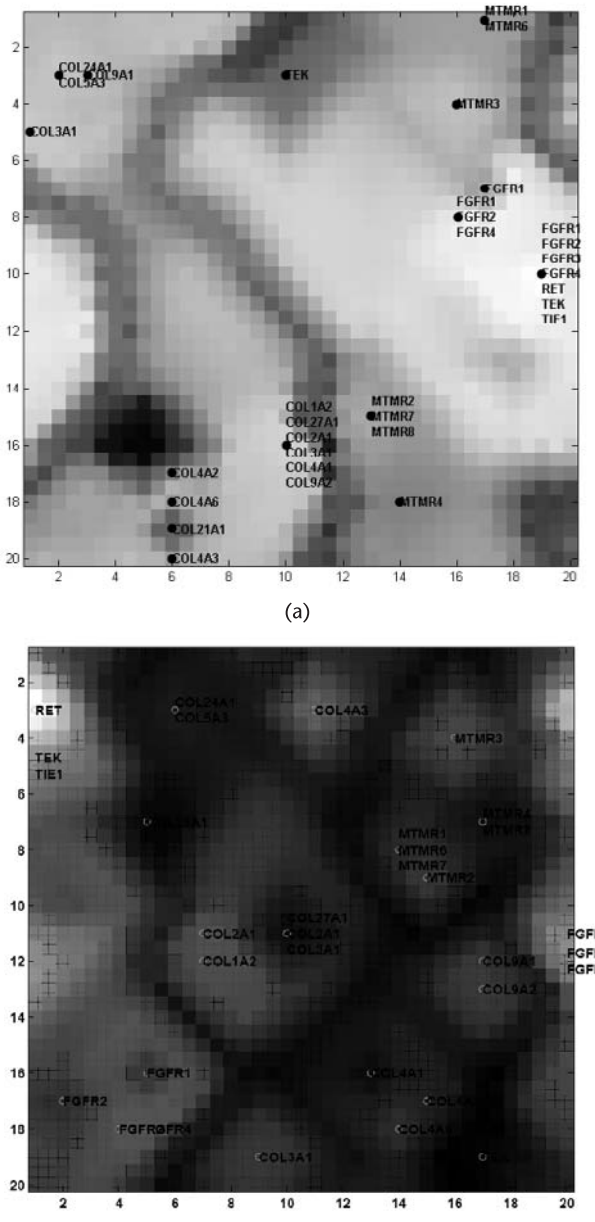


Figure 3.6 The OSOFM map (a) and standard SOM map (b) for the GPD194 dataset.

3.6 Conclusion

In this chapter, we presented several algorithms that use ontologies. NERFCM, a fuzzy relational clustering algorithm, can be used to cluster objects described by ontology terms. The dissimilarity between objects can be computed as in Chapter 2, but also with other distance measures that can deal with multiple variable types

Table 3.2 Most Representative Terms of the OSOM Network Shown in Figure 3.6

OSOM Index	*GO ID*	*GO Definition*
M:(17, 1)	GO:0006470	Protein amino acid dephosphorylation
C:(2, 3)	GO:0005201	Extracellular matrix structural constituent
R:(10, 3)	GO:0016301	Kinase activity
M:(16, 4)	GO:0006470	Protein amino acid dephosphorylation
C:(1, 5)	GO:0005201	Extracellular matrix structural constituent
R:(17, 7)	GO:0004872	Receptor activity
R:(16, 8)	GO:0004713	Protein-tyrosine kinase activity
R:(19, 10)	GO:0005524	ATP binding
M:(13, 15)	GO:0006470	Protein amino acid dephosphorylation
C:(10, 16)	GO:0005201	Extracellular matrix structural constituent
C:(6, 17)	GO:0005201	Extracellular matrix structural constituent
C:(6, 18)	GO:0005201	Extracellular matrix structural constituent
M:(14, 18)	GO:0006470	Protein amino acid dephosphorylation
C:(6, 19)	GO:0007155	Cell adhesion
C:(6, 20)	GO:0005587	Collagen type IV

Note: (M)-myotubularin, (R)-receptor precursor, (C)-collagen

(see examples in [8, 38]). The resulting fuzzy cluster memberships can be used in automatic ontology annotation based on the guilt-by-association paradigm or in data summarization (see [27, 29] and Chapter 8 for more examples). Related to NERFCM, we presented CCV, a cluster-validity measure for relational datasets. It, too, can be used in data summarization.

Last, we presented OSOM, a version of the well-known self-organizing maps (SOM) algorithm, that was modified to include Gene Ontology term-dissimilarity information.

We believe that the inclusion of ontological information in existent clustering algorithms can lead to new knowledge-discovery tools that are able to reveal new facets of the represented objects.

References

[1] Altschul, S. F., et al., "Basic Local Alignment Search Tool," *J Mol Biol*, Vol. 215, No. 3, 1990, pp. 403–410.

[2] Bellazzi, R., and B. Zupan, "Towards Knowledge-Based Gene Expression Data Mining," *J. of Biomedical Informatics*, Vol. 40, 2007, 787–802.

[3] Ben-dor, A., and Z. Yakhini, "Clustering Gene Expression Patterns," *J. of Computational Biology*, Vol. 6, 1999, pp. 281–297.

[4] Bezdek, J. C., *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York: Plenum, 1981, p. 272.

[5] Bezdek, J. C., and R. J. Hathaway, "VAT: A Tool for Visual Assessment of (Cluster) Tendency," *Proc. IJCNN 2002*, HI, May 12–17, 2002, pp. 2225–2230.

[6] Bezdek, J.C., et al., *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, Boston: Springer, 1999, p. 796.

[7] Bolshakova, N., F. Azuaje, and P. Cunningham, "A Knowledge-Driven Approach to Cluster Validity Assessment," *Bioinformatics*, Vol. 21, No. 10, 2005, pp. 2546–2547.

[8] Boriah, S., V. Chandola, and V. Kumar, "Similarity Measures for Categorical Data: A Comparative Evaluation," Siam 2008, pp 243–254.

[9] Enright, A. J., S. Van Dongen, and C. A. Ouzounis, "An Efficient Algorithm for Large-Scale Detection of Protein Families," *Nucleic Acids Res*, Vol. 30, No. 7, 2002, pp. 1575–1584.

[10] Frey, B.J., and D. Dueck, "Clustering by Passing Messages Between Data Points," *Science* Vol. 315, No. 972, 2007.

[11] Handl, J., J. Knowles, and D. B. Kell, "Computational Cluster Validation in Post-Genomic Data Analysis," *Bioinformatics*, Vol. 21, No. 15, 2005, pp. 3201–3212.

[12] Hathaway, R. J., and J. C. Bezdek, "NERF c-Means: Non-Euclidean Relational Fuzzy Clustering," *Pattern Recognition*, Vol. 27, No. 3, 1994, pp. 429–437.

[13] Havens, T.C., et al., "Ontological Self-Organizing Maps for Cluster Visualization and Functional Summarization of Gene Products Using Gene Ontology dissimilarity Measures," *World Congress on Computational Intelligence, WCCI2008*, Hong Kong, June, 1–6, 2008, pp. 104–109.

[14] Henegar, C., et al., "Clustering Biological Annotations and Gene Expression Data to Identify Putatively Co-Regulated Biological Processes," *J. Bioinf. Comp. Biol.*, Vol. 4, No. 4, August 2006, pp. 833–52.

[15] Huang, D., and W. Pan, "Incorporating Biological Knowledge into Distance Based Clustering Analysis of Microarray Gene Expression Data," *Bioinformatics*, Vol. 22, 2006, pp. 1259–1268.

[16] Kaski, S., and T. Kohonen, "Exploratory Data Analysis by the Self-Organizing Map: Structures of Welfare and Poverty in the World," *Neural Networks in Financial Engineering, Proc., 3rd Int. Conf. on Neural Networks in the Capital Markets*, P. N. Refenes, et al., (eds.), London, Singapore: World Scientific, 1996, pp. 498–507.

[17] Klir, G. J., and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Upper Saddle River, New Jersey: Prentice Hall, 1995, p. 574.

[18] Kohonen, T., "Self-Organized Formation of Topologically Correct Feature Maps," *Biological Cybernetics*, Vol. 43, 1982, pp. 59–69,

[19] Kohonen, T., "Self-Organizing Maps," *Proc. IEEE*, Vol. 78, No. 9, September 1990, pp. 1464–1480.

[20] Kohonen, T., "Self-Organizing Maps," *Information Sciences*, Vol. 30, 2004.

[21] Kustra, R., and A. Zagdanski, "Incorporating Gene Ontology in Clustering Gene Expression Data," *19th IEEE Symp. on Computer-Based Medical Systems, IEEE Computer Society*, 2006, pp. 555–563.

[22] Myllyharju, J., and K. Kivirikko, "Collagens, Modifying Enzymes, and Their Mutation in Humans, Flies, and Worms," *Trends in Genetics*, Vol. 20, No. 1, 2004, pp. 33–43.

[23] Martin, D., et al., "GOToolBox: Functional Analysis of Gene Datasets Based on Gene Ontology," *Genome Biol.* Vol. 5, No. 12, 2004, p. R101.

[24] Melton, G. B,, et al., "Inter-Patient Distance Metrics Using SNOMED CT Defining Relationships," *J. of Biomedical Informatics*, Vol. 39, No. 6, December 2006, pp. 697–705.

[25] Pal, N. et al., "Gene Ontology-Based Knowledge Discovery Through Fuzzy Cluster Analysis," *Neural, Parallel and Scientific Computation*, Vol. 13, Nos. 3–4, 2005, pp. 337–361.

[26] Pedersen, T., et al., "Measures of Semantic Dissimilarity and Relatedness in the Biomedical Domain," *J. of Biomedical Informatics*, Vol. 40, 2007, pp. 288–299.

[27] Popescu, M., et al., "Functional Summarization of Gene Product Clusters Using Gene Ontology Dissimilarity Measures," in *Proc. 2004 ISSNIP*, M. Palaniswami, et al., (eds.), Piscataway, New Jersey: IEEE Press, 2004, pp. 553–559.

[28] Popescu, M., and J. Arthur, "OntoQuest: A Physician Decision Support System Based on Ontological Queries of the Hospital Database," *Proc. AMIA Fall Symp.*, Washington, D.C., November 2006, pp. 639–643.

[29] Popescu, M., and J. Keller, "Summarization of Patient Groups Using the Fuzzy C-Means and ICD-9 Ontology Dissimilarity Measures," *IEEE World Congress on Computational Intelligence*, Vancouver, Canada, July 16–21, 2006, pp. 2998–3003.

[30] Popescu, M., J. M. Keller, and J. A. Mitchell, "Fuzzy Measures on the Gene Ontology for Gene Product Dissimilarity," *IEEE Trans. Computational Biology and Bioinformatics*, Vol. 3, No. 3, July–September 2006, pp. 1–11.

[31] Popescu, M., et al., "A New Cluster Validity Measure for Bioinformatics Relational Datasets," *World Congress on Computational Intelligence, WCCI2008*, Hong Kong, June 1–6, 2008, pp. 726–731.

[32] Runkler, T., "Relational Fuzzy Clustering," *Advances in Fuzzy Clustering and Its Applications*, J. V. de Oliveira and W. Pedrycz (eds.), New York: John Wiley & Sons, Ltd., 2007, pp. 31–51.

[33] Schlicker, A, et al., "A New Measure for Functional Dissimilarity of Gene Products Based on Gene Ontology," *BMC Bioinformatics*, Vol. 7, No. 302, 2006, pp 1–16.

[34] Speer, N., C. Spieth, and A. Zell, "Functional Grouping of Genes Using Spectral Clustering and Gene Ontology," in *Proc. of the IEEE Int. Joint Conf. on Neural Networks (IJCNN 2005)*, Piscataway, New Jersey: IEEE Press, 2005, pp. 298–303.

[35] Speer, N., C. Spieth, and A. Zell, "A Memetic Clustering Algorithm for the Functional Partition of Genes Based on the Gene Ontology," *Proc. of the 2004 IEEE Symp. on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2004)*, Piscataway, New Jersey: IEEE Press, 2004, pp. 252–259.

[36] Wang, H., et al., "Gene Expression Correlation and Gene Ontology-Based Dissimilarity: An Assessment of Quantitative Relationships," *Proc. of the 2004 IEEE Symp. on Computational Intelligence in Bioinformatics and Computational Biology*, La Jolla, CA, October 7–8, 2004, Piscataway, New Jersey: IEEE Press, 2004, pp. 25–31.

[37] Wang, J.Z., et al., "A New Method to Measure the Semantic Dissimilarity of GO Terms," *Bioinformatics*, Vol. 23, No. 10, 2007, pp. 1274–1281.

[38] Wilson, R, and T. Martinez, "Improved Heterogeneous Distance Functions," *JAIR*, Vol. 6, 1997, pp. 1–34.

CHAPTER 4

Analyzing and Classifying Protein Family Data Using OWL Reasoning

Katy Wolstencroft, Rachel Brenchley, Lydia Tabernero, and Robert Stevens

4.1 Introduction

The classification of the genes and proteins expressed by an organism is an important step in understanding its molecular biology. Much of this process can be automated by applying bioinformatics tools to the sequence data. Genes can be predicted, and the functions of the resulting proteins can be characterized by similarity searching and domain-architecture analysis. These analyses, however, describe the sequence features of proteins, but they do not classify them. This is often where the automation stops. Expert curators perform the final step of classification. Scientific curators can recognize the functional properties that are sufficient to place an individual gene product into a particular protein family group. Automating this final classification step would be advantageous in order to manage the growing number of genomes and the rapid changes in knowledge about protein families.

This chapter describes the use of description-logic reasoning over an OWL ontology to automate the classification of proteins into family and subfamily groups. The ontology captures the domain-architecture properties of the different members of a protein family. Protein instances can be analyzed using standard sequence-analysis tools, and a description-logic reasoner can classify them as instances of particular classes by the combinations of domains they contain.

This is a novel approach for applying ontology reasoning to biological data. Instead of using the ontology simply as a static vocabulary for annotation, we use the formal class descriptions in OWL to classify and catalog data in an analysis pipeline.

We demonstrate the automated classification system using a large protein family, the protein phosphatases. Studies of the human and Aspergillus fumigatus genomes found that our knowledge-based, automatic classification matches, and sometimes surpasses, that of the human curators. We have made the classification process fast and reproducible, and where appropriate knowledge is available, the method can be generalized for use with any protein family. This methodology does

not use any new bioinformatics techniques or algorithms for detecting sequence features. Instead, it augments existing tools by providing a novel method for interpreting the results of these techniques and algorithms to perform automatic protein classification.

The final part of the chapter describes the use of this classification system for a comparative study of the protein phosphatases from three parasite species, *Trypanosoma brucei, Trypanosoma cruzi,* and *Leishmania major.* This study is the first application of the ontology classification method in the field, illustrating the advantages of such an approach. The work described in this chapter has been published in [1–3].

4.1.1 Analyzing Sequence Data

This method focuses on classifying proteins into families and subfamilies. A protein family is a group of proteins, descended from a common ancestor, typically sharing similar functions and three-dimensional structures, as well as sequence similarity.

In order to classify proteins in this way, we need to first analyze the raw sequence data to identify which sequences are protein phosphatases, and within those sequences, which phosphatase family and subfamily groups they belong to.

There are several methodologies which could be employed to discover and extract protein phosphatase sequences using bioinformatics tools. For instance, similarity searching with a tool such as BLAST [4] could be used to identify all sequences over a certain similarity threshold, but this alone does not determine the differences between subfamily members. The demarcation between closely-related subfamilies would remain ambiguous. An additional approach would be to analyze the domain architecture of each protein.

Many proteins are assemblies of functional domains and/or motifs (hereafter referred to as p-domains for *protein domains*). Each p-domain might have a separate function within the protein, such as catalysis or regulation, but it is the composition of the different p-domains that gives each protein its specific function. Performing this type of analysis would enable small functional differences to be determined, but these tools simply report the presence of domains, not the consequences of the p-domain combinations for classification.

There are many tools dedicated to discovering functional p-domains in proteins, for example, PROSITE [5], SMART [6], and Pfam [7]. Each tool employs different methods. PROSITE uses pattern matching to detect single motifs and domains, whereas Pfam uses hidden Markov models (HMMs). The InterProScan tool [8] provides an integrated view of these and many other functional domain resources. InterProScan combines all of the different techniques, allowing all to be accessed from a single query.

In certain cases, the presence of a p-domain is diagnostic for membership in a particular protein family, for example, the protein tyrosine kinase's catalytic domain is diagnostic of the tyrosine kinases. Classification at a fine-grained level, into subfamilies, however, is not usually possible without further analysis. For automated classification methods, this need for extra human intervention limits performance. Ontologies provide a technology for capturing and using this human understanding of an area of research within computer applications.

4.1.2 The Protein Phosphatase Family

Protein phosphatases, in conjunction with protein kinases, are involved in the control and regulation of numerous biological processes and cellular pathways. For example, cell signaling cascades, cell cycle regulation, homeostasis, and cell growth and differentiation [9, 10]. It is estimated that in a eukaryotic genome, approximately 3% of expressed genes encode protein kinases or protein phosphatases [11], and that, at any one time, one-third of proteins in eukaryotic cells are phosphorylated [12], demonstrating the importance and abundance of these molecules in cellular function.

The implication of phosphatases in human diseases, such as diabetes, cancer, and neurodegenerative conditions [13–15], makes the protein phosphatase family an interesting target for medical and pharmaceutical research. The size of the family means that classification at a detailed level is vital for understanding the biological role of individual proteins and for comparative genomic studies. Phosphorylation events have also been found to be important for controlling the life cycle of parasites, which was the motivation for applying this analysis method to the study of phosphatases in the three newly sequenced parasite genomes.

The phosphatase group of enzymes are divided into four distinct gene families, PPP and PPM [16], which are both serine/threonine phosphatase families, PTP [17, 18], which are protein tyrosine phosphatases and protein histidine phosphatases [19]. Additionally, the lipid phosphatases [20] are often included as part of the PTP gene superfamily. In vivo, their physiological substrates are phosphoinositides, but they have been shown to exhibit poor catalytic activity with protein substrates in vitro, and there is a distinct evolutionary relationship with the PTP family [21]. Dual specificity phosphatases (DUSP) are also included in the PTP superfamily, and they can dephopsphorylate both phosphotyrosine and phosphoserine/threonine residues [18].

Recent reviews on the protein phosphatase family [17, 18, 22, 23] focus on either tyrosine phosphatases or serine/threonine phosphatases. There have been extensive studies into the characterization of each in the human genome. Although each type of phosphatase performs the same chemical reaction in the cell, the removal of a phosphate group, there are distinct differences in their biological roles and catalytic specificity [16].

Most serine/threonine phosphatases are multisubunit complexes, combining a catalytic subunit with regulatory and targeting subunits. The final combination of subunits produces the resulting number of each serine/threonine phosphatase in a given organism. For example, the protein phosphatase 1 catalytic subunit binds to different regulatory subunits. Approximately 100 of these regulatory subunits have been identified to date [24], providing differences in substrate specificity, subcellular localization, and enzymatic activity.

The tyrosine phosphatase family presents a less-complicated picture. Instead of protein complexes, they are single polypeptides with different noncatalytic domains providing differences in specificity or subcellular and tissue location. The necessity for fine-grained classification is, however, increased with the subtlety of the differences between closely related proteins performing different functions.

4.2 Methods

4.2.1 The Phosphatase Classification Pipeline

Membership of the PPP, PPM, PTP, and the lipid phosphatases can be determined by the presence of phosphatase catalytic domains. This enabled the optimization of the steps involved in gathering and annotating the raw data. Analyzing a protein using InterProScan can take several minutes, but since we were only interested in proteins with a phosphatase catalytic domain, we could use other screening methods to filter these before proceeding with InterProScan. The whole process was further optimized by designing a Taverna workflow [25] to automate the orchestration between the prescreening filter, the InterProScan analysis, and the loading of the raw data into the ontology reasoning system. Taverna is a workflow-management system, which allows the interconnection of distributed services and data resources.

The following steps describe the workflow of bioinformatics processes performed in this application:

1. Prescreen for protein phosphatase sequences from among all protein sequences, using the Web-service interface to the EMBOSS program, patmatdb [26];
2. InterProScan on each protein phosphatase to determine its domain composition, using the InterProScan Web service available from the EBI;[1]
3. Transformation of the XML output from the InterProScan analysis into protein instances for the OWL ontology, using a Web-service wrapper of a bespoke local Java script;
4. Reasoning over the phosphatase ontology to infer to which class of protein phosphatase each protein instance belongs.

The final step in this process usually requires human analysis, but in this method, it can be supported computationally by the use of an ontology. A protein phosphatase ontology, expressed in OWL, captures the necessary and sufficient properties for membership in each protein phosphatase subfamily. The reasoner can compare these properties with the domain-architecture properties of each protein phosphatase instance, as determined by the preceding data analysis steps.

4.2.2 The Datasets

The human protein phosphatases have been extensively studied and classified by experts in the field. The domain architectures of each phosphatase family and subfamily have been well characterized, and the differences between them have been described (see Figure 4.1) [18]. The availability of detailed knowledge makes this group of proteins a suitable test case for the ontology reasoning methodology. Demonstrating that reasoning over the ontology can classify the human proteins in the same way as human experts validates the methodology. To demonstrate the generic applicability of this method, we also used it to analyze proteins from Aspergillus fumigatus. At the time of the study, the A. fumigatus genome had recently

1. http://www.ebi.ac.uk/Tools/webservices/services/InterProScan

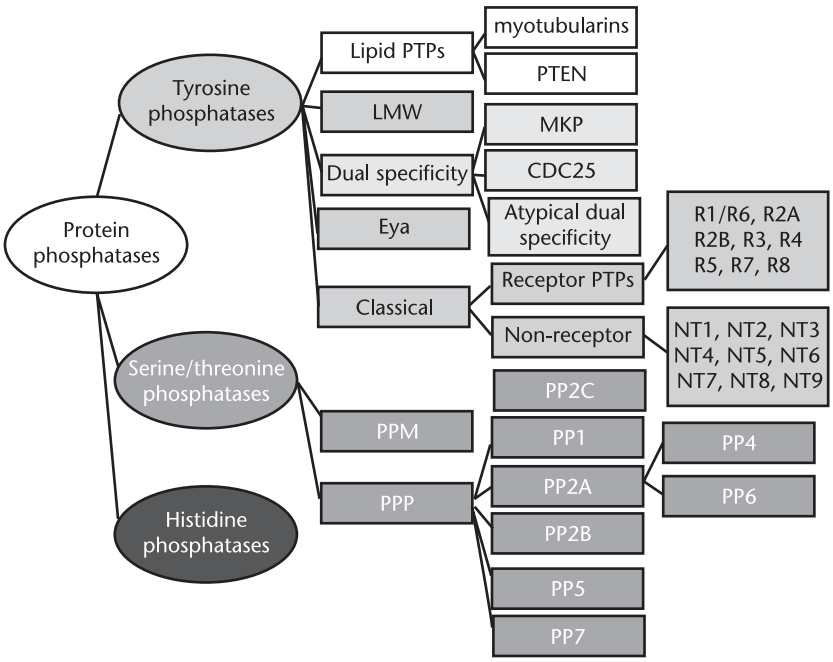


Figure 4.1 Relationships between the different family and subfamily groups in the protein phosphatase family.

been sequenced and the data pertaining to protein sequences had been analyzed only by automated-sequence analysis methods, with very little curation. Therefore, this data enabled a comparison of the automated classification with that of simple automated-sequence analysis. Finally, the ontology method was used to analyze three closely-related protozoan parasite genomes: *Trypanosoma brucei, Trypanosoma cruzi,* and *Leishmania major.* These were new datasets on which no previous analysis had been performed. In this case, the objective was to identify, catalog, and compare the protein phosphatases from each organism for studies into cell signaling, which has been found to play a part in infection by these parasites. It was expected that the three parasite genomes would have similar numbers and types of phosphatises, due to their close evolutionary relationships, and that we would see a more divergent picture between the human, Aspergillus, and parasite data.

4.2.3 The Phosphatase Ontology

The ontology is a representation of expert knowledge in the area of protein phosphatases. It is a model that describes the properties of each family and subfamily and the physical p-domain features required for inclusion in each family and subfamily. It contains descriptions of each major protein phosphatase family, the PTP, PPM, PPP, and lipid phosphatases, with a set of necessary and sufficient properties (conditions) for membership in each. These major families are further divided into subfamilies, and additional properties for each subsequent subfamily are also

described. This creates a hierarchy with any subfamily group inheriting properties from the more general family classes. The open-world assumption, which underpins the OWL language, is essential for the classification to work correctly. The same is true for the use of disjoint axioms and also qualified cardinality.

For example, a tyrosine phosphatase *must* contain a tyrosine phosphatase catalytic domain, and a serine/threonine phosphatase *must* contain a serine/threonine catalytic domain. Not only must the domains be present, but their presence is enough to recognize any given protein as a member of the class in question. These are the *necessary and sufficient* conditions for membership in one of the main family groups. A protein is an instance of the tyrosine phosphatase family if it contains *at least one* PTP catalytic domain. This is the diagnostic property for this protein family.

The OWL axioms say what must be present, but do not indicate that these domains are the only ones present; there may be others. It simply has not, as yet, been stated. If there are other domains present, they may enable further classification into subfamilies, or they may simply be other properties of a protein sequence that do not feature in the classification. The open-world assumption allows for this style of description. A class definition is constructed from a collection of properties and axioms that are necessary and sufficient to distinguish it from other classes, but an individual may have more properties besides these, providing it has these as a minimum.

Disjoint axioms are equally useful for classification and distinguishing closely related individuals. For example, the defined classes tyrosine phosphatase and serine/threonine phosphatase are sibling classes and are disjoint. This means that protein sequences can satisfy only one of these sets of properties and be placed as an individual in only one of these classes. Disjointness does not need to be asserted between defined classes, as the necessary and sufficient criteria enable the reasoner to work out to which classes any instance belongs, or which defined class might subsume another. Disjointness can also be asserted between primitive classes. Such axioms help describe to which classes an instance belongs.

The segregation of the protein phosphatases at the level of tyrosine verses serine/threonine reflects the current knowledge of the domain. There are currently no examples of phosphatases that contain both catalytic domains. If we were to find data to suggest otherwise, we would have to refine the ontology model.

The same disjoint axiom pattern is used throughout the ontology, allowing the reasoner to differentiate and classify instances between sibling class assignments. Qualified cardinality restrictions are applied in areas in which two sibling class definitions contain the same combination of p-domains, but in different quantities.

Figure 4.2 illustrates the p-domain architectures of the PTP family, demonstrating that each can be differentiated by cataloging and counting the presence and absence of p-domains. Figures 4.3(a) and 4.3(b) provide examples of class definitions from the ontology.

The full protein phosphatase ontology is available at: http://www.bioinf.manchester.ac.uk/phosphabase/links.html. The ontology describes classes of phosphatases, but not individual proteins. An Instance Store [27] was used in order to reason over the descriptions of individual proteins and to enable the storage of those descriptions. Instances can be asserted and stored in OWL ontologies (i.e.,

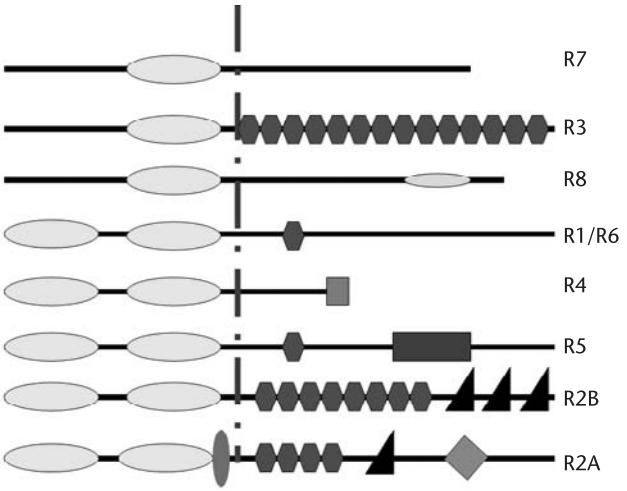


Figure 4.2 The differences in domain architecture of the receptor tyrosine phosphatase subfamily. White oval = phosphatase catalytic domain. Vertical bar = transmembrane region, white triangle = immunoglobulin domain, black hexagon = fibronectin domain, black diamond = MAM domain, black rectangle = carbonic anhydrase domain, grey oval = adhesion recognition site, white square = glycosylation, and black oval = cadherinlike domain

(Class Protein Sequence type Tyrosine Phosphatase

– (contains at Lease 1 Protein Tyrosine Phosphatase Domain) and
– (contains 1 Transmembrane Domain))

(a)

(Class Protein sequence type R2A (receptor tyrosine phophatase 2A)

–(contains 2 protein tyrosine phosphatase domains) and
–(contains 1 transmembrane domain) and
–(contains 4 fibronectine domains) and
–(contains 1 immunoglobulin domain) and
–(contains 1 MAM domain) and
–(contains 1 cadherin-like domain))

(b)

Figure 4.3 (a) Necessary and sufficient properties for membership in the protein tyrosine phosphatase family and (b) necessary and sufficient properties for membership in the receptor tyrosine phosphatase 2A subfamily. R2A is a subclass of tyrosine phosphatase, so the R2A definition, therefore, also fulfills the conditions for membership in the tyrosine phosphatase family.

inside Protégé[2]), but at the start of this work there was a potential problem with scalability. Adding more than approximately 1,000 instances affected the performance of reasoning over the data. It was not expected that individual genomes would typically contain more than this number of phosphatases, but to make use of such a technology in comparative studies, it was a consideration.

2. http://www.co-ode.org/downloads/protege-x/

These problems have now been overcome with a combination of improvements in description-logic reasoners, such as Pellet,[3] and improvements in ontology-editing tools, such as Protégé. Protégé 4 is more scalable, so the use of the Instance Store may no longer be required for this methodology.

The Instance Store combines a description-logic reasoner with a relational database. The OWL ontology is loaded into the Instance Store, and the reasoner uses this to perform the task of classification; that is, from the OWL instance descriptions given, it determines the appropriate ontology class for an instance description. The relational database provides the stability, scalability, and persistence necessary for this work. The Instance Store itself provides a relatively simple programmatic interface, allowing the assertion of descriptions and queries against the set of instances. It uses highly optimized algorithms to denormalize datasets as they are asserted and later determine whether the information in the database is sufficient to answer queries, or whether reasoning is required.

4.3 Results

4.3.1 Protein Phosphatases in Humans

The human phosphatase classification results validated the ontology model and the methodology. The ontology classification matched that of the human experts for each of the 118 proteins analyzed. A detailed comparison of the automatic classification and the human expert classification can be found in [2].

There were additional benefits from using the new method. The results provided an opportunity to refine the classification further. In two of the dual-specificity phosphatases, the study identified additional functional domains [2].

In [18], Alonso et al., describe the atypical dual-specificity phosphatases as being divided into seven subtypes. The largest of these have the same domain architecture; they contain tyrosine phosphatase and dual-specificity catalytic domains. However, several proteins have additional functional domains that have been shown to confer functional specificity [28]. Classifying the proteins using the ontology highlighted more of these extra domains.

The protein DUSC contains a zinc-finger domain (IPR007087). This protein has been characterized not only in the human genome [29], but in many other species [30]. In the classification presented by Alonso et al. in [18], the protein is present, but is wrongly annotated as containing a FYVE domain. FYVE domains are different types of zinc-finger domains that occur in the myotubularin proteins, MTMR3 and MTMR4. Earlier reviews of the tyrosine phosphatase family, however, do include the zinc-finger domain in the protein [31]. These results illustrate an inconsistency in the accepted protein phosphatase community knowledge and highlight a possible disadvantage of human-expert annotation, namely, human error.

The dual-specificity phosphatase 10 protein (DUSP10) contains a disintegrin domain. Its UniProt record reflects this,[4] but the domain does not appear in any

3. http://clarkparsia.com/pellet/
4. http://www.uniprot.org/uniprot/Q9Y6W6

phosphatase characterization or classification study. The domain architecture of DUSP10 is conserved in other species, which suggests a specific function for the domain, but currently available experimental evidence does not explain what this might be.

4.3.2 Results from the Analysis of *A. Fumigatus*

In contrast to the human proteins, the phosphatases from Aspergillus fumigatus are not well characterized. At the time of the study, the genome sequence had only been available for a short period of time, and classification and annotation were underway through the Central Aspergillus Data Repository (CADRE).[5]

The function of some proteins had been determined by experimental methods, but most were simply predicted from *in silico* methods, by which functions were inferred using automated similarity searches. The annotations of the proteins with inferred functions reflected this, with terms such as *hypothetical* and *putative* as part of the description. Therefore, in this case, the ontology method was being used as a primary method of classification. The results were compared to those obtained using automated similarity searches, and it was found that the ontology method provided more detailed classification.

The analysis results also provided a foundation for studying the differences between the protein phosphatases expressed in humans and in *A. fumigatus*. Figure 4.4 shows the comparative abundance of the family and subfamilies of phosphatases in *A. fumigatus* and human genomes.

The protein serine/threonine phosphatase composition remains relatively unchanged, but there are radical differences between the tyrosine and dual-specificity subfamilies. Firstly, the number of proteins in *A. fumigatus* is greatly reduced. Where the human genome contains 16 myotubularin proteins and 11 MAP kinase phosphatase proteins, *A. fumigatus* contains only one of each. The number of classical protein tyrosine phosphatases is also reduced. There are no incidences of non-receptor tyrosine phosphatases and only three receptor tyrosine phosphatases.

When analyzing these results, the complexity of the two organisms must be taken into account. *A. fumigatus* is a pathogenic mold, and as such, protein phosphatases with tissue-specific expression in humans [32], for example, would not be expected to be conserved.

The ontology classification uncovered an *A. fumigatus* protein phosphatase with a novel domain architecture that was not present in the human phosphatases. Protein Afu5g09360 is a calcineurin protein (PP2B) that contains an extra homeobox domain. The homeobox domain binds to DNA using a helix-turn-helix structural motif. It is found in a variety of DNA-binding proteins, many of which are transcription factors.

PP2B is well conserved throughout evolution. Performing BLAST analyses on Afu5g09360 (data not shown) and InterProScans of the proteins exhibiting the most similarity revealed that the homeobox domain in PP2B was present in other Aspergillus species and closely related fungi, but was not present in any other taxonomic group. This conservation strongly suggests a specific function for this extra

5. http://www.cadre-genomes.org.uk/

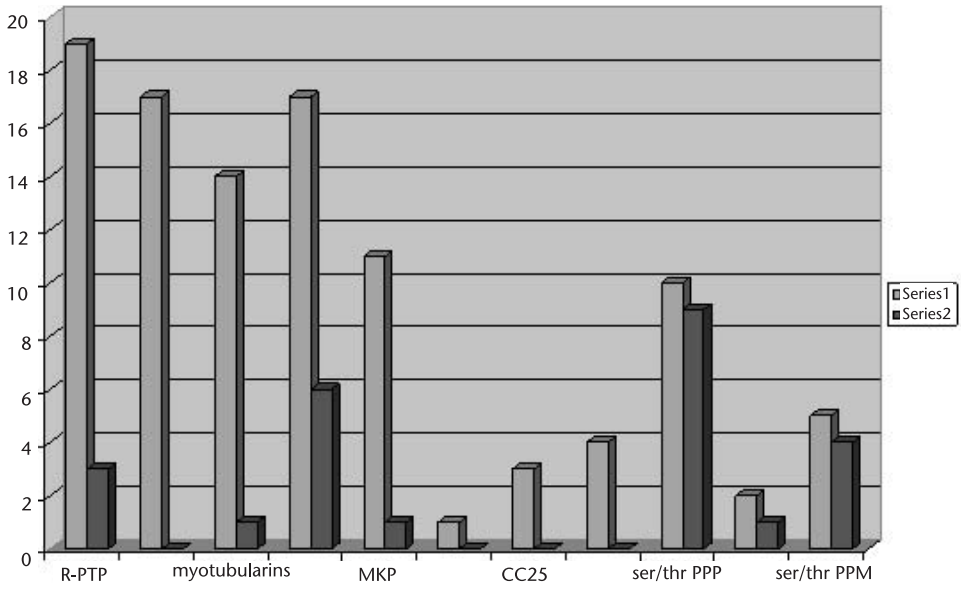


Figure 4.4 The numbers and subfamily types of protein phosphatases in human and *A. fumigatus* species. Human data is shown in gray and *A. fumigatus* in black.

domain. Previous studies have identified a divergence in the mechanisms of action of calcineurin in pathogenic fungi [33] and have also demonstrated that this is critical for virulence. Other studies on one function of calcineurin in Arabidopsis, Na^+ ion homeostasis, [34] have revealed that a homeobox protein, Athb-12, is also involved. This study raises the possibility of a similar regulatory role for the homeobox domain in the *A. fumigatus* protein, but laboratory experimentation is required to investigate this possibility.

4.3.3 Ontology System Versus *A. Fumigatus* Automated Annotation Pipeline

Comparing the ontology classification method with that of automated similarity search prediction methods yielded promising results. In many cases, the automated prediction approach underperformed when compared to the ontology system. The ontology classification placed proteins into more specific classes. For example, the ontology classified the protein Afu1g05640 as a myotubularin, a specific subclass of the dual-specificity phosphatases, which is a lipid phosphatase. The annotation from the *A. fumigatus* sequencing consortium simply stated that it was a protein phosphatase (Table 4.1).

There was one case in which the automated similarity search appeared to provide a more detailed classification than the ontology, but with further investigation, this was proven false. The protein Afu2g11990 was annotated as a Pten phosphatase, whereas the ontology simply classified it as a dual-specificity phosphatase (the parent class of Pten). On closer inspection, however, the protein did not contain

Table 4.1 A Comparison of the Differences in Classification Between the Automated Annotations Assigned to Phosphatases by the *A. Fumigatus* Sequencing Project and Classification by the Ontology Method

A. Fumigatus Annotation		*Ontology Classification*
Afu1g03540	Hypothetical protein	Dual-specificity phosphatase
Afu1g05640	Protein phosphatase	Myotubularin
Afu5g11690	Related to protein tyrosine phosphatase PPS1	Dual-specificity phosphatase
Afu4g07080	Putative dual-specificity phosphatase	Dual-specificity phosphatase
Afu4g07000	Tyrosine phosphatase	Tyrosine phosphatase
Afu4g04710	Putative tyrosine phosphatase	MAP kinase phosphatase (MKP)
Afu6g06650	Conserved hypothetical protein	Tyrosine phosphatase
Afu2g11990	Pten-3-phosphoinisitide phosphatase	Dual-specificity phosphatase
Afu3g12250	Putative protein tyrosine phosphatase	Dual-specificity phosphatase
Afu2g02760	Putative protein tyrosine phosphatase	Dual-specificity phosphatase
Afu3g10970	Protein tyrosine phosphatase	Protein tyrosine phosphatase
Afu1g04950	Serine/threonine protein phosphatase 1	Classical serine/threonine phosphatase
Afu1g09280	Protein phosphatase 2C putative	Protein phosphatase 2C
Afu1g15800	Protein phosphatase 2C putative	Protein phosphatase 2C

domains indicative of Pten proteins (18). A sequence similarity search revealed partial similarity to the Pten protein from Dictyostelium discoideum, but this was in the region of the dual-specificity phosphatase domain, so there does not appear to be sufficient evidence to place this protein in the Pten phosphatase class.[6]

Overclassification is as much of a problem as underclassification. In time, the *A. fumigatus* data will be compared to related genomes, or inferences will be made about these proteins based on potentially misleading annotations.

This demonstrates a clear advantage of the ontology approach. The ontology classification is based on physical evidence from analyzing the sequence data. If functional domains are not detected, they cannot form part of the classification, and the protein becomes an instance of a less-specific class.

A detailed description of the *A. fumigatus* phosphatase classification, before and after ontology classification, can be found in Wolstencroft et al. [1, 2], but Table 4.1 provides a summary.

4.4 Ontology Classification in the Comparative Analysis of Three Protozoan Parasites—A Case Study

As unicellular organisms, kinetoplastid parasites have little in common with humans and other metazoans, except for essential eukaryotic cell processes preserved through evolution. To further determine the applicability of the ontology method

6. Since this work, the PTEN protein subfamily has been studied in more detail and clearer diagnostic domains have been identified. Under this classification, there is more evidence that the protein in question is a PTEN protein.

for protein phosphatase classification across all organisms, three protozoan parasites were analysed, *Trypanosoma brucei, Trypanosoma cruzi,* and *Leishmania major.* The results from this analysis were also biologically useful in their own right. The similarities and differences in cellular signaling mechanisms between these organisms are important in the study of the human diseases for which they are responsible (the Tritryps diseases).

An important reason for choosing these organisms was the fact that their protein kinases (kinomes) had also been analyzed using domain architecture and evolutionary analyses [35], which allows more insight to be gained into the overall signaling processes of lesser-studied lower eukaryotes, for both phosphorylation and dephosphorylation. The protein phosphatase analysis was, therefore, essential to fully understand cell signaling.

4.4.1 TriTryps Diseases

T. brucei, T. cruzi, and *L. major* are the causative agents of three human diseases endemic in many third-world countries: African sleeping sickness, Chagas' disease, and cutaneous Leishmaniasis, respectively. These are vector-borne diseases, transmitted to humans, and sometimes to animals, by the bite of an infected Tsetse fly, assassin bug, or sand fly. The genome sequences for these three organisms have been published [36–38], and they have collectively become known as the TriTryps. Recently, two other *Leishmania* species, *Leishmania braziliensis* and *Leishmania infantum* [39], have also been sequenced, and their genomes are available at the GeneDB database (http://www.genedb.org).

Millions of people are at risk from African sleeping sickness, Chagas' disease, and cutaneous Leishmaniasis, and several factors prevent an easy eradication. There is limited access to healthcare for large numbers of people, as it is often remote, poor areas where the diseases are prevalent. In addition, *T. brucei* infects cattle, which has a huge impact on rural communities and implications for struggling third-world economies (http://www.who.int/mediacentre/factsheets/fs259/en/). Drugs are available to treat sleeping sickness and Chagas' disease, but for sleeping sickness, this depends on diagnosis at an early stage. Even then, some drugs have serious and often toxic side effects. It is thought unlikely that a vaccine can be synthesised for Chagas' disease as *T. cruzi* has been found to initiate an autoimmune response [40]. A vaccine against cutaneous Leishmaniasis, sandfly saliva proteins, and others from Leishmania species has been found to be potentially useful [41].

4.4.2 TriTryps Protein Phosphatases

Phosphorylation events have been found to be important for controlling the life cycle of these parasites. The *T. brucei* protein *Tb*PTP1 is a protein tyrosine phosphatase and a master regulator of life-cycle differentiation in the parasite [42]. Efforts to synthesize effective vaccines and therapies depend on a detailed understanding of the signaling pathways of kinetoplastid parasites, which are modified throughout different stages of their life cycles. Protein phosphatases are an important part of this mechanism, and it is important to know when different phosphatases are ac-

tive, how they are regulated, and what their substrates are to be able to develop useful antiparasitic drugs.

4.4.3 Methods for the Protozoan Parasites

The method for this investigation was identical to the human and Aspergillus studies, except that the prescreening stage was omitted. All proteins were analyzed using InterProScan, and protein phosphatases were identified from the InterProScan results. InterProScan is much more sensitive than a simple pattern-matching search, and this precaution was taken to ensure that no potential phosphatases were missed from the analysis. There is no evidence from the human and *A.fumigatus* studies that this would happen, but in those studies we had the expert-curated sequences and automated annotation for comparison. For the parasite genomes, we had no such data available. Omitting the prescreening step also demonstrated the extensibility of the method. It is not necessary to confine studies to narrowly defined groups of uniform proteins if the initial results from InterProScan can discriminate between the higher levels of protein families.

The prescreening stage was originally included as a measure for reducing computational time. Therefore, the main disadvantage of omitting this step is the increase in computational time required for the InterProScan analysis of thousands or tens of thousands of sequences, rather than a few hundred. Approximately 40,000 sequences in total were submitted to InterProScan for the three genomes. The results of this analysis (i.e., the protein instances) were classified using the ontology.

4.4.4 Sequence Analysis Results from the TriTryps Phosphatome Study

In total, 250 proteins with phosphatase catalytic domains were extracted and classified from the three organisms [3]. Through detailed examination of these sequences, domain architectures, key conserved sequence motifs, and evolutionary relationships were determined. Comparisons between the three species and also with humans provided interesting results.

At the time of the investigation, there had been no comprehensive analysis of protein phosphatases in these parasitic organisms. Numerous experimental studies had identified a few important tyrosine phosphatases, but most work had been done on the PPP type of serine/threonine phosphatases. These include PP1 and PP2A [43] and PP5 [44] in *T. brucei,* PP1 in *T. cruzi* [45], and PP7/PPEF in all three [46].

The ontology classification results showed that more than one-third of the total protein phosphatases are atypical phosphatases in these kinetoplastids [3]. Atypical refers to those sequences in which any of the following apply:

1. The domain organization is novel;
2. There is no homologue in higher eukaryotes;
3. Motifs usually conserved in eukaryotic catalytic domains are not present or contain significant differences that may affect the structure and function of the protein.

The high number of atypical phosphatases suggests that the original model was highly biased towards the higher eukaryotes. Upon inspection of InterPro, it was found that some domains and motifs had not been updated recently, and so they did not include sequences from more recently analyzed genomes. For example, the domain for cdc25 DSPs, IPR000751 (not used in this analysis), is partly based on a fingerprint (PR00716) from the PRINTS database [47], and this entry was last updated in 1999. It was generated using only mammalian, fruit fly (*Drosophila melanogaster*), pig (*Sus scrofa*), and African clawed frog (*Xenopus laevis*) sequences. This can create bias toward the identification of phosphatases from the higher eukaryotes and limit the use of the pattern in finding sequences in more distantly related species. If more organisms were represented in the InterPro domain models, this would improve the efficiency of phosphatase identification for lower eukaryotes.

Atypical dual-specificity phosphatases (DSPs) form a large proportion of the total number of atypical proteins in the TriTryps. There are three cdc14 sequences that are truly similar to higher eukaryotes; the remainder have unusual, unique domain combinations or simply lack any great similarity to metazoan phosphatases. Figure 4.5 shows the relative abundance of atypical and eukaryoticlike phosphatases in each of the three organisms.

An exciting discovery in the TriTryp analysis was the existence of three atypical sequences containing a DSP catalytic domain at the C-terminal, leucine-rich repeats (LRR), and two inactive N-terminal protein kinase domains. The presence of both a phosphatase catalytic domain and a kinase catalytic domain (active or inactive) within the same protein sequence is rare. This type of sequence would presumably be very useful, biologically, as substrates could be phosphorylated and dephosphorylated by the same protein. For organisms with small genomes, this may be an advantage. These unique proteins were named kinatases from *kin*ase and phosph*atase*). There is much evidence that inactive pseudokinases can still play important roles in cell signaling [48], and so this could have novel functions in protozoa.

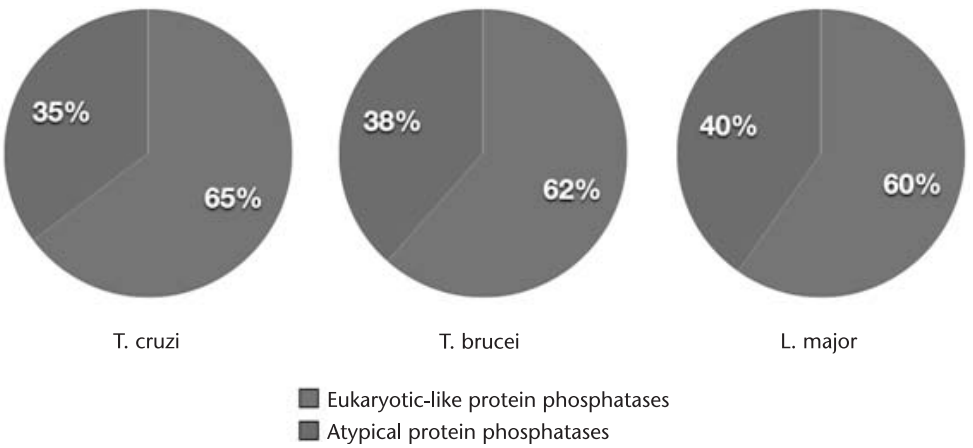


Figure 4.5 The relative abundance of eukaryoticlike and atypical protein phosphatases in three protozoan parasites.

Comparisons between the three protozoans reveal important biological differences. *T. cruzi* and *L. major* are both intracellular parasites, whereas *T. brucei* resides in the bloodstream. *L. major* invades macrophages, and *T. cruzi* will invade many different cell types, including macrophages and fibroblasts. There will be important differences in cell biochemistry, allowing parasites to exist in these different environments in the human body. Although they have great similarity, with regard to the numbers of sequences in each phosphatase subtype, the classification study showed several proteins to be present in the intracellular parasites and not in *T. brucei*. Firstly, *T. brucei* has no eukaryoticlike PTEN sequence. There are two types of PTEN in these parasites. The first is similar to the human sequence, possessing a PTEN catalytic domain and a C-terminal lipid-binding domain, which is thought to assist the catalytic domain by binding to the cell membrane in a productive location [49]. *T cruzi* and *L. major* each possess one eukaryoticlike sequence each. *T. brucei,* however, has only the atypical type, which contains a catalytic domain alone and no C-terminal region. *T. brucei* is also missing the serine/threonine phosphatase PP6 and does not possess an arsenate reductase sequence with similarity to cdc25 DSPs, as do *T. cruzi* and *L. major.*

4.4.5 Evaluation of the Ontology Classification Method

In total, the ontology extracted and classified 142 protein phosphatases from the three parasite species. For the majority of sequences (approx 78%), the ontology classification provided more information than their original annotations, or the same level of detail. Therefore, the ontology system again surpassed, or was equal to, the automated sequence analysis method in the majority of cases. The remainder of the sequences suffered a loss of information, being classified as a member of the PTP, PPM, or PPP parent classes, instead of in individual subfamily groups, or were false positives (i.e., proteins that did not belong to the protein phosphatase family).

There were 10 false-positive sequences that had been placed as members of the general tyrosine phosphatase family (7 sequences) and low molecular weight PTPs (LMW-PTPs) (3 sequences). A multiple alignment had to be performed to determine the lack of conserved motifs in these sequences. For the LMW-PTPs, the InterPro domain is not specific enough to distinguish between LMW-PTPs and other very similar types of enzymes, such as arsenate reductases for the lower eukaryotes [50]. If the prescreening method had been employed before this analysis, however, these false positives would not have been detected. Again, this problem and the loss of information from a small number of sequences can be largely attributed to the higher eukaryotic bias in some of the InterPro domains.

The ontology classification method was as good as or better than large-scale automated sequence annotation methods, supplying a fast analysis of the phosphatase gene products and providing a good deal of information that was not previously known about kinetoplastid parasite phosphatases. To increase the efficiency of the system, however, we must address the issues of bias towards higher eukaryotes in InterPro for some sequences. One way to do this is to combine this analysis with other bioinformatics tools before the classification stage. The workflow for extracting and initially analyzing data could be extended to accomplish this task.

4.5 Conclusion

Postgenomic bioinformatics presents new problems for the bioinformatician. The scale of data production has increased dramatically, while the pace of data analysis, annotation, and curation has not kept pace. Often, compromises on the quality of annotation have to be made in order to interpret large datasets quickly. By designing a system that will allow rapid, automated classification to the fine-grained, subfamily level, the necessity to make such a compromise is avoided. This study demonstrates the advantages of combining community knowledge, in an ontology, with automated annotation methods.

Standard automated methods of annotation provide evidence for similarity to other known proteins, or provide lists of functional domains within a protein, but they do not allow the interpretation of this information. The strength of human-expert annotation is in this interpretation step. In a novel approach, the interpretation step was replaced with further automation. Using the technologies of formal description logics and ontological reasoning, community knowledge can be captured and utilized for data analysis. The methodology does, however, hinge on the expert knowledge of a domain. If the data does not fit the current knowledge of a particular area, this is also a useful outcome. It informs the scientists that their model needs revision or expansion.

The ontology system classified the human protein phosphatases with equal competence as human experts, enabling confidence to be placed in similar studies of the protein phosphatases of uncharacterized genomes. This was demonstrated by the results from the parasite genome analysis. It was also discovered that the ontology system was efficient at uncovering novel, unexpected functional domains. As the ontology classified proteins according to what was already known, proteins exhibiting a different composition of domains were easily highlighted, identifying new targets for further scientific research.

This work focused on classifying proteins into family groups using domain-architecture analysis. There are many tools that can be employed in such a task, either instead of InterProScan, or in addition to it. In order to use more data sources and analysis tools in this investigation, we would simply have to extend the workflow that extracts and analyses the data before the ontology classification. In fact, this methodology is applicable in any area of biology in which class membership can be defined according to a set of properties that can be derived using automated analysis tools. To date, the use of ontological technology in biology has been largely restricted to enhancing browsing and querying over existing data. Harnessing the reasoning capabilities of DL ontologies in this way to enable automated classification could potentially have a great impact on bioinformatics analyses and approaches to automation in the future.

As well as extending the data-collection part of the ontology classification process, we can also increase the expressivity of the protein class descriptions. For example, for the protein phosphatases, the order of p-domains was not important, but simply counting the number of each was sufficient to distinguish between proteins from different subfamilies. In other protein families, however, the order of the p-domains would also need to be specified. If we take the ABC transporters

as an example, the ABCD and ABCG subfamilies have exactly the same p-domain architecture. The only difference is the orientation of their two p-domains, an ATP-binding domain and a transmembrane region. ABCG proteins are referred to as *reverse* transporters, as the ATP-binding domain is N-terminal to the transmembrane domain, which is the opposite orientation to the ABCD proteins.

Ontology use in the bioinformatics community is continuing to grow, providing data-management solutions and the ability to define concepts and terms across large, disparate research communities. Despite these advances, however, the full use of the reasoning capabilities of formal DL ontologies is not being exploited in many cases. The automated protein classification using the ontology reasoning presented here demonstrates the extra advantages of using these capabilities. It is hoped that this system can be employed and exploited in future work, for example, in drug-target identification and new genome annotation.

References

[1] Wolstencroft, K., et al., "A Little Semantic Web Goes a Long Way in Biology," *4th Int. Semantic Web Conf.*, Vol. 3792, Galway, Ireland, November, 6–10, 2005, pp. 786–800.

[2] Wolstencroft, K., et al., 2006. "Protein Classification Using Ontology Classification," *Bioinformatics,* Vol. 22, 2006, pp. e530–e538.

[3] Brenchley, R., et al., "The TriTryp phosphatome: Analysis of the Protein Phosphatase Catalytic Domains," *BMC Genomics*, Vol. 8, No. 434, 2007.

[4] Altschul, S. F., et al., "Gapped BLAST and PSI-BLAST: A New Generation of Protein Database Search Programs," *Nucleic Acids Res.*, Vol. 25, 1997, pp. 3389–3402.

[5] Hulo, N., et al., (2004) "Recent Improvements to the PROSITE Database," *Nucleic Acids Res.*, Vol. 32, 2004, pp. D134–D137.

[6] Letunic, I., et al.,"SMART 4.0: Towards Genomic Data Integration," *Nucleic Acids Res.*, Vol. 32, 2004.

[7] Finn, R.D. et al., "The Pfam Protein Families Database," *Nucleic Acids Res.*,Vol. 36 (Database Issue), January 2008, pp. D281-D288, Epub. November 26, 2007.

[8] Hunter, S., et al., "InterPro: The Integrative Protein Signature Database," *Nucleic Acids Res.*, Vol. 37 (Database Issue), January 2009, pp. D211-D215, Epub. October 21, 2008.

[9] Barford, D., A. K. Das, and M. P. Egloff, "The Structure and Mechanism of Protein Phosphatases: Insights into Catalysis and Regulation," *Annu. Rev. Biophys Biomol. Struct.*, Vol. 27, 1998, pp. 133–164, Review.

[10] Tonks, N. K., and B. G. Neel, "Combinatorial Control of the Specificity of Protein Tyrosine Phosphatases," *Curr Opin Cell Biol*, Vol. 13, 2004, pp.182–195.

[11] Depaoli-Roach, A. A., et al., "Serine/Threonine Protein Phosphatases in the Control of Cell Function," *Adv Enzyme Regul*, Vol. 34, 1994, pp.199–224.

[12] Zolnierowicz, S., and M. Bollen, "Protein Phosphorylation and Protein Phosphatases," De Panne, Belgium, September 19–24, 1999, *Embo J,* Vol. 19, 2000, pp. 483–488.

[13] Schonthal, A. H., "Role of Serine/Threonine Protein Phosphatase 2A in Cancer," *Cancer Lett.*, Vol. 170, No. 1, 2001, pp. 1–13.

[14] Zhang, Z. Y., "Protein Tyrosine Phosphatases: Prospects for Therapeutics," *Curr. Opin. Chem. Biol.*, Vol. 5, No. 4, 2001, pp. 416–423.

[15] Tian, Q., and J. Wang, "Role of Serine/Threonine Protein Phosphatase in Alzheimer's Disease," *Neurosignals*, Vol. 11, No. 5, 2002, pp. 262–269.

[16] Barford, D., A. K. Das, and M. P. Egloff, "The Structure and Mechanism of Protein Phosphatases: Insights into Catalysis and Regulation," *Annu. Rev. Biophys. Biomol. Struct.*, Vol. 27, 1998, pp. 133–164, Review.

[17] Andersen, J. N., et al., "A Genomic Perspective on Protein Tyrosine Phosphatases: Gene Structure, Pseudogenes, and Genetic Disease Linkage," *FASEB J.*, Vol. 18, No. 1, 2004, pp. 8–30, Review.

[18] Alonso, A., et al., "Protein Tyrosine Phosphatases in the Human Genome," *Cell*, Vol. 117, No. 6, 2004, pp. 699–711, Review.

[19] Klumpp, S., et al., "Protein Histidine Phosphatase: A Novel Enzyme with Potency for Neuronal Signaling," *J Cereb Blood Flow Metab*, Vol. 22, 2002, pp. 1420–1424.

[20] Nandurkar, H. H., and R. Huysmans, 2002. "The Myotubularin Family: Novel Phosphoinositide Regulators," *IUBMB Life*,Vol. 53, 2002, pp. 37–43.

[21] Wishart, M. J., and J. E. Dixon, "PTEN and Myotubularin Phosphatases: From 3-Phosphoinositide Dephosphorylation to Disease," *Trends Cell Biol*, Vol. 12, 2002, pp. 579–585.

[22] Andersen, J. N., et al., "A Genomic Perspective on Protein Tyrosine Phosphatases: Gene Structure, Pseudogenes, and Genetic Disease Linkage," *FASEB J.*, Vol. 18, No. 1, 2004, pp. 8–30, Review.

[23] Cohen, P., "Novel Protein Serine/Threonine Phosphatases: Variety is the Spice of Life," *Trends Biochem. Sci.*, Vol. 22, No. 7, 1997, pp. 245–51, Review.

[24] Bollen, M., and W. Stalmans, "The Structure, Role, and Regulation of Type 1 Protein Phosphatases," *Crit Rev Biochem Mol Biol*, Vol. 27, 1992, pp. 227–281.

[25] Oinn, T., et al., "Taverna: A Tool for the Composition and Enactment of Bioinformatics Workflows," *Bioinformatics,* Vol. 20, No. 17, 2004, pp. 3045–3054.

[26] Rice, P., I. Longden, and A. Bleasby,"EMBOSS: The European Molecular Biology Open Software Suite," *Trends in Genetics*, Vol. 16, No. 6, 2000, pp. 276–277.

[27] Bechhofer, S., I. Horrocks, and D. Turi, "Implementing the Instance Store," *Computer Science*, preprint CSPP-29, University of Manchester, 2004.

[28] Wang, J., et al., "A Unique Carbohydrate Binding Domain Targets the Lafora Disease Phosphatase to Glycogen," *J. Biol. Chem.*, Vol. 77, No. 4, 2002, pp. 2377–2380.

[29] International Human Genome Sequencing Consortium, "Finishing the Euchromatic Sequence of the Human Genome," *Nature,* Vol. 431, No. 7011, 2004, pp. 931–945.

[30] Kumar, R., et al., "A Zinc-Binding Dual-Specificity YVH1 Phosphatase in the Malaria Parasite, Plasmodium falciparum, and Its Interaction with the Nuclear Protein, Pescadillo," *Mol Biochem Parasitol.*, Vol. 133, No. 2, 2004, pp. 297–310.

[31] Bhaduri,A., and R. Sowdhamini, "A Genome-Wide Survey of Human Tyrosine Phosphatases," *Protein Eng*, Vol. 16, No. 12, 2003, pp. 881–888.

[32] Chagnon, M. J., N. Uetani, and M. L. Tremblay, "Functional Significance of the LAR Receptor Protein Tyrosine Phosphatase Family in Development and Diseases," *Biochem. Cell Biol.*, Vol. 82, No. 6, 2004, pp. 664–675.

[33] Kraus, P. R., and J. Heitman, "Coping with Stress: Calmodulin and Calcineurin in Model and Pathogenic Fungi," *Biochemical and Biophysical Research Communications*, Vol. 311, 2003, pp. 1151–1157.

[34] Shin, D., et al., "Athb-12, a Homeobox-Leucine Zipper Domain Protein from Arabidopsis Thaliana, Increases Salt Tolerance in Yeast by Regulating Sodium Exclusion," *Biochem Biophys Res Commun*, Vol. 323, 2004, pp. 534–540.

[35] Parsons, M., et al., "Comparative Analysis of the Kinomes of Three Pathogenic Trypanosomatids: Leishmania Major, Trypanosoma Brucei and Trypanosoma Cruzi," *BMC Genomics*, Vol. 6, 2005, p. 127.

[36] Berriman, M., et al., "The Genome of the African Trypanosome Trypanosoma Brucei," *Science*, Vol. 309, No. 5733, 2005, pp. 416–422.

[37] El-Sayed, N. M., et al., "The Genome Sequence of Trypanosoma Cruzi, Etiologic Agent of Chagas Disease," *Science*, Vol. 309, No. 5733, 2005, pp. 409–415.

[38] Ivens, A. C., et al., (2005), "The Genome of the Kinetoplastid Parasite, Leishmania Major," *Science*, Vol. 309, No. 5733, 2005, pp. 436–442.

[39] Peacock, C. S., et al., "Comparative Genomic Analysis of Three Leishmania Species That Cause Diverse Human Disease," *Nat Genet*, Vol. 39, No. 7, 2007, pp. 839–847.

[40] Rose, N. R., "Infection, Mimics, and Autoimmune Disease," *J. Clin. Invest*, Vol. 107, No. 8, 2001, pp. 943–944.

[41] Reithinger, R., et al., "Cutaneous Leishmaniasis," *Lancet. Infect Dis.*, Vol. 7, No. 9, 2007, pp. 581–596.

[42] Szoor, B., et al., "Protein Tyrosine Phosphatase TbPTP1: A Molecular Switch Controlling Life Cycle Differentiation in Trypanosomes," *J Cell Biol*, Vol. 175, No. 2, 2006, pp. 293–303.

[43] Erondu, N. E., and J. E. Donelson, "Characterization of Trypanosome Protein Phosphatase 1 and 2A Catalytic Subunits," *Mol Biochem Parasitol*, Vol. 49, No. 2, 1991, pp. 303–314.

[44] Chaudhuri, M., "Cloning and Characterization of a Novel Serine/Threonine Protein Phosphatase Type 5 from Trypanosoma Brucei," *Gene*, Vol. 266, Nos. 1–2, 2004, pp. 1–13.

[45] Orr, G. A., et al., "Identification of Novel Serine/Threonine Protein Phosphatases in Trypanosoma Cruzi: A Potential Role in Control of Cytokinesis and Morphology," *Infect Immun*, Vol. 68, No. 3, 2000, pp. 1350–1358.

[46] Mills, E., et al., "Kinetoplastid PPEF Phosphatases: Dual Acylated Proteins Expressed in the Endomembrane System of Leishmania," *Mol Biochem Parasitol*, Vol. 152, No. 1, 2007, pp. 22–34.

[47] Attwood, T. K., "The PRINTS Database: A Resource for Identification of Protein Families," *Brief Bioinform*, Vol. 3, No. 3, 2002, pp. 252–263.

[48] Boudeau, J., et al., "Emerging Roles of Pseudokinases," *Trends Cell Biol*, Vol. 16, No. 9, 2006, pp. 443–452.

[49] Lee, J. O., et al., "Crystal Structure of the PTEN Tumor Suppressor: Implications for Its Phosphoinositide Phosphatase Activity and Membrane Association," *Cell*, Vol. 99, No. 3, 1999, pp. 323–334.

[50] Mukhopadhyay, R., and B. P. Rosen, "Arsenate Reductases in Prokaryotes and Eukaryotes," *Environ Health Perspect*, Vol. 110, Suppl 5,, 2002, pp. 745–748.

CHAPTER 5

GO-Based Gene Function and Network Characterization

Gyan Prakash Srivastava, Trupti Joshi, Zhao Song, Chao Zhang, Guan Ning Lin, Ping Li, James Andrew Ross, Mihail Popescu, Jingdong Liu, Jing Qiu, and Dong Xu

5.1 Introduction

Despite considerable efforts, a large number of genes in most organisms have not been functionally annotated, and a significant number of the current gene annotations contain errors. This fact emphasizes the challenging nature of the problem and the need for accurate function prediction. Gene function prediction can be extraordinarily valuable to researchers for functional inference and follow-up experimental design [1, 2]. Numerous computational methods have been developed over the past years to either use a particular type of data or integrate different data sources for characterizing gene-gene associations and predicting gene functions [1, 3–8]. Nevertheless, the state of the art of gene-function prediction has significant room for improvement [1]. Hence, gene function prediction is still an active research area.

Using an ontology is essential for gene-function prediction. The most widely used ontology for gene-function prediction is GO (Gene Ontology, see Chapter 1 for details), which has been broadly recognized as the most comprehensive classification system for gene functions in modern biology [9]. GO provides a controlled vocabulary to describe gene and gene-product attributes in any organism. It is important to have an ontology such as GO in order to conduct large-scale biological studies in the postgenomic era. Before GO was developed, gene functions were annotated using natural language, which may be inconsistent and ambiguous and is hard to use in computations. GO provides a resource for handling gene function systematically. With GO, it is easy to develop computational methods for gene-function prediction.

In this chapter, we will introduce various methods of integrating high-throughput data for gene-function prediction based on GO. The basic idea for our gene-prediction approach is to use high-throughput data (sequence, gene expression, protein interaction, etc.) to establish a relationship between a query gene and genes with known functions. For this purpose, we will first define GO-based measures of functional relationships in Section 5.2. Then we will demonstrate how GO-based functional relationships among genes can be revealed in high-throughput data in

Section 5.3. We will introduce the theories on how to establish functional relationships among genes based on high-throughput data in Section 5.4. Section 5.5 will discuss the algorithms of GO-based gene-function prediction, using functional relationships derived from high-throughput data. In Section 5.6, we will show some tests and application examples of gene-function predictions. Section 5.7 extends gene-function prediction to gene-network studies. Section 5.8 discusses the related software that we developed. We will have additional discussions in Section 5.9.

5.2 GO-Based Functional Similarity

Chapter 2 of this book has extensive discussions on ontology-based similarity measures. Such similarity measures are particularly important for gene-function prediction, as they quantify the functional relationship and provide a basis for inferring the function of a query gene according to its relationships to genes of known functions. In this section, we introduce two similarity measures specifically related to our studies.

5.2.1 GO Index-Based Functional Similarity

The graph structure of GO can be utilized to compute similarities between two GO terms. We used a numerical GO index to represent the structure of each of three categories separately. The deepest level in the index is 18. A GO index, as denoted by a string of linked numbers, for example, 1-4-2-29, characterizes the GO annotation of every protein. The first number corresponds to the type of ontology annotation category, for example, 1 represents biological process, 2 represents molecular function, and 3 represents cellular component. An example of a GO index for the cell-growth process and related functions is described below.

1-4 Cell growth and/or maintenance, GO:0008151
1-4-3 Cell cycle, GO:0007049
1-4-3-2 DNA replication and chromosome cycle, GO:0000067
1-4-3-2-4 DNA replication, GO:0006260
1-4-3-2-4-2 DNA dependent DNA replication, GO:0006261
1-4-3-2-4-2-2 DNA ligation, GO:0006266

For example, consider a gene pair, ORF1 and ORF2, both of which are annotated with GO functions. Assume ORF1 has a function represented by GO index 1-1-3-3-4 and ORF2 has a function represented by 1-1-3-2. When compared with each other for the level of matching GO indices, they match through index level 1 (1-1) and level 2 (1-1-3), and will have functional similarity equal to 2. Functional similarity defined this way can assume values from 1 to 18.

5.2.2 GO Semantic Similarity

We can also calculate functional similarity between two GO terms in terms of semantic similarity (see Chapter 2). An example of calculating the semantic similarity is shown in Figure 5.1.

To calculate semantic similarity between two genes, the probability of each GO term assigned to a gene is derived first. For each gene in an organism, the probability is calculated by counting the number of the descendants of an assigned GO term plus 1 (the GO term itself), divided by the total number of GO-term annotations in the organism. The probability of each node increases as we go toward the root of the GO ontology, which is defined as biological process (GO:0008150), molecular function (GO:0003674), or cellular component (GO:0005575) in the three ontologies. The semantic similarity between two ontology terms, t_1 and t_2, is defined as

$$SS(t_1, t_2) = -\ln p_{ms}(t_1, t_2) \tag{5.1}$$

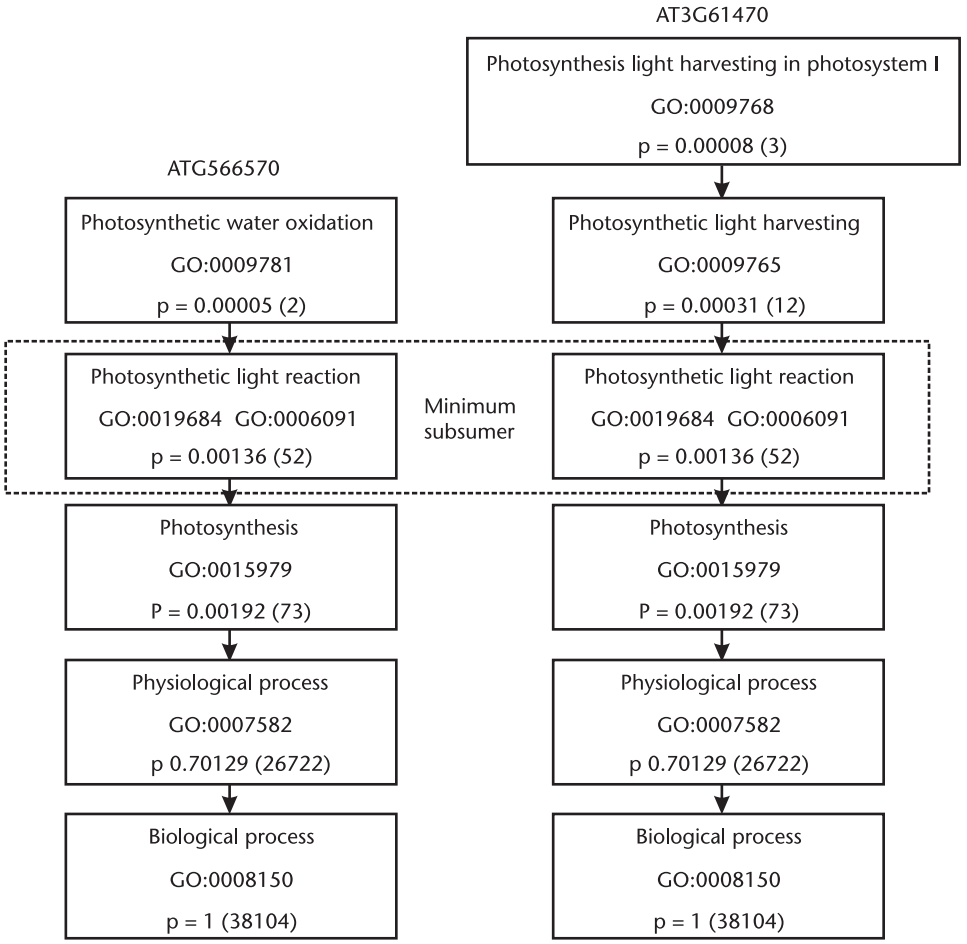


Figure 5.1 Example of semantic similarity between two GO terms.

where $P_{ms}(t_1,t_2)$ is the relative frequency of the minimum subsumer for terms t_1 and t_2. The minimum subsumer for terms t_1 and t_2 is defined as the common parent of the deepest GO-index level shared by t_1 and t_2.

5.3 Functional Relationship and High-Throughput Data

Several computational methods exist for predicting gene functions using relevant high-throughput data [2, 6, 10–20]. However, none of these methods perform well enough for broad biological applications. There are various reasons for this. Clearly, the algorithms of these methods can be improved. Another potential reason is that the underlying relationship between gene-gene functional similarity and various kinds of high-throughput data is not well characterized. In this section, we will demonstrate some simple relationships between GO-based functional similarity of a gene pair and different high-throughput data using the examples of microarray data and gene sequence data.

5.3.1 Gene-Gene Relationship Revealed in Microarray Data

One of the classical approaches to exploring the connection between the functional similarities of a gene pair and their associated microarray expression profiles includes calculating the standard Pearson correlation coefficient between the profiles. Our early studies [10, 11, 14] have shown that at higher correlations, the associated two genes tend to have more similarity in terms of their functions. Figure 5.2 shows a higher probability of sharing the same function for broader functional categories, as expected. It also shows the probability of a gene pair sharing the same level of

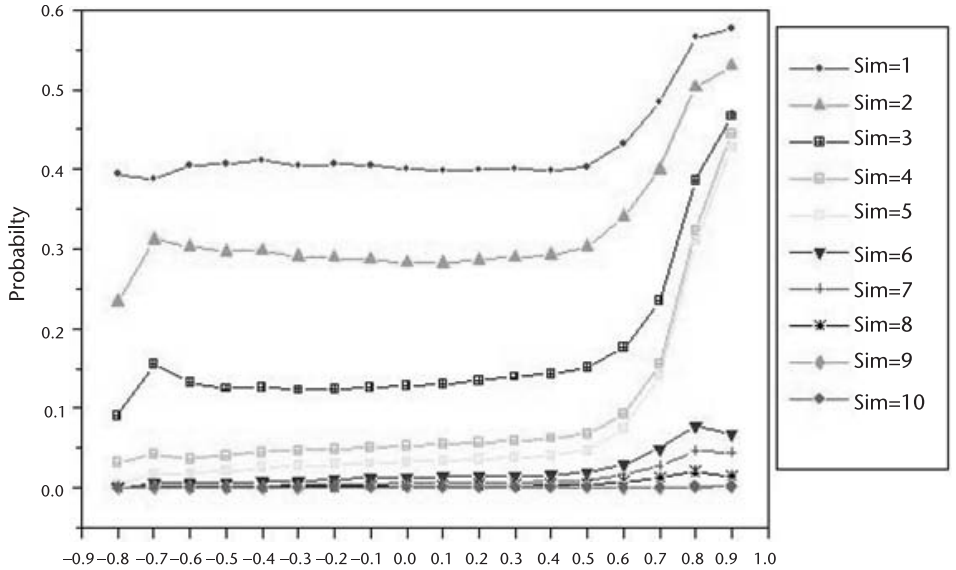


Figure 5.2 Probabilities of gene pairs sharing the same levels of GO indices against Pearson correlation coefficient of microarray gene-expression profiles. The Pearson correlation coefficient ranges from –1 to 1.

GO-indices increases versus the Pearson correlation coefficient of their microarray gene-expression profiles. Clearly, there exists a relationship between the correlation in microarray gene-expression profiles and similarity in gene function.

5.3.2 The Relation Between Functional and Sequence Similarity

Various sequence-function relationships were identified by measuring the correlation between sequence identity (or expectation value) and GO-index similarity (or semantic similarity) within the same genome or across different genomes for the three GO categories [21]. Figure 5.3 shows a consistent correlation between the functional similarity of the biological-process ontology at different GO-index levels and the expectation values (E-values) of sequence alignment using BLAST [22]. There is also a higher functional similarity for the lower GO-index levels. This is mainly due to the fact that there is a higher chance for two randomly picked genes to share the GO index at the lower level. Meanwhile, the functional similarity consistently increases as the E-value decreases, given the same GO-index level.

5.4 Theoretical Basis for Building Relationship Among Genes Through Data

Section 5.3 demonstrated that there is rich information contained in high-throughput data to characterize functional relationships among genes and to predict gene function. In this section, we will discuss how to formulate such relationships using statistical methods and computational algorithms.

5.4.1 Building the Relationship Among Genes Using One Dataset

We used four different types of high-throughput data and quantified the pairwise distance between two genes by using different distance measures for different types of data, as shown in Table 5.1.

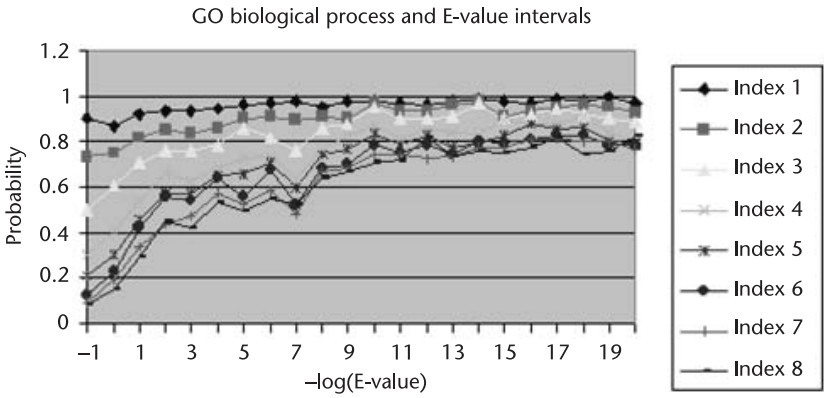


Figure 5.3 The probability for two similar genes to share the same function at a particular GO index versus the negative logarithmic (base 10) E-value of sequence similarity based on BLAST within the same genomes using the GO biological-process annotations.

Table 5.1 High-Throughput Data and Measurement of Gene Relationship

Data Type	*Method*
Gene expression	Pearson correlation coefficient
Protein-protein interaction	Binary (0,1)
Phylogenetic profile	Pearson correlation coefficient
Protein domain	Maryland-Bridge coefficient

In the following, we will discuss the method to characterize the gene-gene relationship from each data type.

5.4.1.1 Gene Expression

Gene expression is recorded as an $n \times m$ matrix with n genes, each of which has m experimental conditions or time points. We used the Pearson correlation coefficient, r, as the pairwise measure of the linear relationship between two gene profiles. The following equation measures the Pearson correlation between profiles X and Y:

$$r = \frac{m\sum xy - \left(\sum x\right)\left(\sum y\right)}{\sqrt{m\left(\sum x^2\right) - \left(\sum x\right)^2}\sqrt{m\left(\sum y^2\right) - \left(\sum y\right)^2}} \tag{5.2}$$

5.4.1.2 Protein-Protein Interaction

Protein-protein interaction data is recorded as an $n \times n$ matrix I for n genes. If two proteins, i and j, have an interaction, $I_{ij} = 1$; otherwise $I_{ij} = 0$.

5.4.1.3 Phylogenetic Profiles

A phylogenetic profile [23, 24] is a string that encodes the presence or absence of a homologous gene in a set of genomes. It is represented by an $n \times p$ matrix, where n is the number of homologous genes (orthologs) considered, and p is the number of organisms used to generate the profile. The Pearson correlation coefficient is used as a distance measure for phylogenetic profiles.

5.4.1.4 Protein Domains (Pfam and InterPro)

The protein-domain data [25, 26] is represented by an $n \times d$ binary matrix, where n is the number of genes, and d is the number of domains. We calculated the Maryland Bridge distance [27] to characterize the relationship between domain profiles as follows:

$$S_{ab} = \frac{X_{ab}}{2}\left(\frac{1}{X_{aa}} + \frac{1}{X_{bb}}\right) \tag{5.3}$$

where X_i represents the binary vectors of gene i corresponding to the ith row of the matrix, and $X_{ij} = X_i X_j$ is the dot product of two vectors.

5.4.2 Meta-Analysis of Microarray Data

Typically, microarray data is noisy and incomplete. Multiple microarray datasets can be useful for functional inferences in terms of reducing the noise and resulting in a significant addition of sensitivity to extract information from the data. To take advantage of rich information in the different datasets, we combine the statistical meta-analysis [28–31] with our previous gene-function prediction methods [10–12, 14, 27, 32], including other methods to predict gene functions by using multiple microarray datasets.

Because genes that are involved in the same pathway or are part of the same protein complex are often coregulated, a set of genes with similar functions often exhibit expression profiles that are correlated under a large number of diverse conditions or time points [13, 33–35]. As our previous studies showed, there is a significant relationship between functional similarity and Pearson correlation coefficient for a given pair of genes. We evaluated the statistical significance of a Pearson correlation coefficient for two gene-expression profiles in a single dataset, based on the standard *t*-statistics:

$$p - value = P\left(T > \hat{t}\right), \text{ where } \hat{t} = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \tag{5.4}$$

where T is a *t*-random variable with n–2 degrees of freedom, and n is the number of conditions of the gene-expression profiles. Note here that we use the right-tailed p-value, since our previous study [10, 11, 14] and Lee et al. [16] showed that the negative correlation is less likely to be related to functional similarity. A significant p-value for the Pearson correlation implies a coexpression relationship between two genes, and then the function of a gene is predicted according to its coexpressed gene neighbors with known functions. Note that the p-value is monotone with the value of the Pearson correlation coefficient, namely, a large correlation coefficient r implies a small p-value. Therefore, for a single microarray data, choosing coexpressed gene neighbors with a p-value below a threshold is the same as choosing those with the Pearson correlation coefficient above a threshold.

When we have multiple sets of microarray data, the p-values between two genes in single datasets can be combined to obtain the metastatistics, which are then used to establish the functional relationship between the two genes. Since we assume that the datasets are obtained independently, we apply the inverse chi-square method and obtain the metachi-square statistics

$$\widehat{\chi}^2 = \left[-2\log\left(P_1\right) - 2\log\left(P_2\right) - \cdots - 2\log\left(P_n\right)\right] \tag{5.5}$$

where P_i is the p-value obtained from the *i*th dataset for a given gene pair. When there is no linear correlation between a gene pair in any of the multiple datasets, the above chi-square statistic (5.5) $\widehat{\chi}^2$ follows a central chi-square distribution, with degrees of freedom $2n$, and hence, the p-value for meta-analysis, called the *meta p-value*, can be obtained by

$$p_m = P\left(\chi^2_{2n} > \hat{\chi}^2\right) \tag{5.6}$$

where χ^2_{2n} is a chi-square random variable with $2n$ degrees of freedom. For any gene pair of interest, we conclude that the gene pair is positively correlated in at least one of the multiple datasets at level α if the meta p-value is smaller than α. Here, we took a parametric approach to obtain the meta p-value, which is based on the assumption that the distribution of $\hat{t}$ in (5.4) follows a t-distribution with n–2 degrees of freedom under the null hypothesis of no correlation between the gene pair. An examination of the distributions of the observed $\hat{t}$ for all gene pairs and for all datasets showed no obvious departure from this assumption, as shown in Figure 5.4, which plots kernel density (distribution estimate) of the $\hat{t}$ statistics along with the theoretical null density.

When this parametric assumption is a concern, individual p-values can be obtained by comparing the observed t-statistics to the ones generated by randomly permuting the rows within each column Then the meta p-value can be obtained in the same permuted manner as done in [17]. The meta p-value and the Pearson correlation coefficient will be used as the distance measure to calculate the conditional probability that two genes have the same function. This will, in turn, be used for gene-function prediction.

5.4.3 Function Learning from Data

To quantify the gene-function relationship based on a high-throughput data type, we apply Bayes' formula to calculate the conditional probabilities of such gene pairs sharing the same function at each GO-index level, given their distance measure based on the high-throughput data. Here, we use gene-expression data and GO biological-process annotations as an example. For a set of microarray data denoted by M, we use the Pearson correlation coefficient as the distance measure for a single dataset and the meta p-value for multiple datasets [10, 11, 14]. Given a gene pair showing coexpression distance M, the *posterior* probability that two genes share the same function at GO-index level S is

$$p\left(S|M\right) = \frac{p\left(M|S\right)p\left(S\right)}{p\left(M\right)} \tag{5.7}$$

where $p(M|S)$ is the conditional (a priori) probability that two genes are coexpressed in their expression profiles with distance value M, given that two genes have the same GO-index level S. The probability $p(S)$ is the relative frequency that a gene pair has similar functions at the given level of GO-index level, using the annotation data. The probabilities $p(M|S)$ and $p(S)$ are estimated based on the set of genes present in the given dataset of a specific organism whose functions have been annotated with the GO biological processes. The probability $p(M)$ is estimated by the relative frequency of coexpression distance M over all gene pairs in the organism, which is calculated from the genome-wide gene-expression profiles. This conditional probability will be used to predict the set of predicted functions for each query gene from the union of known functions of the neighboring genes.

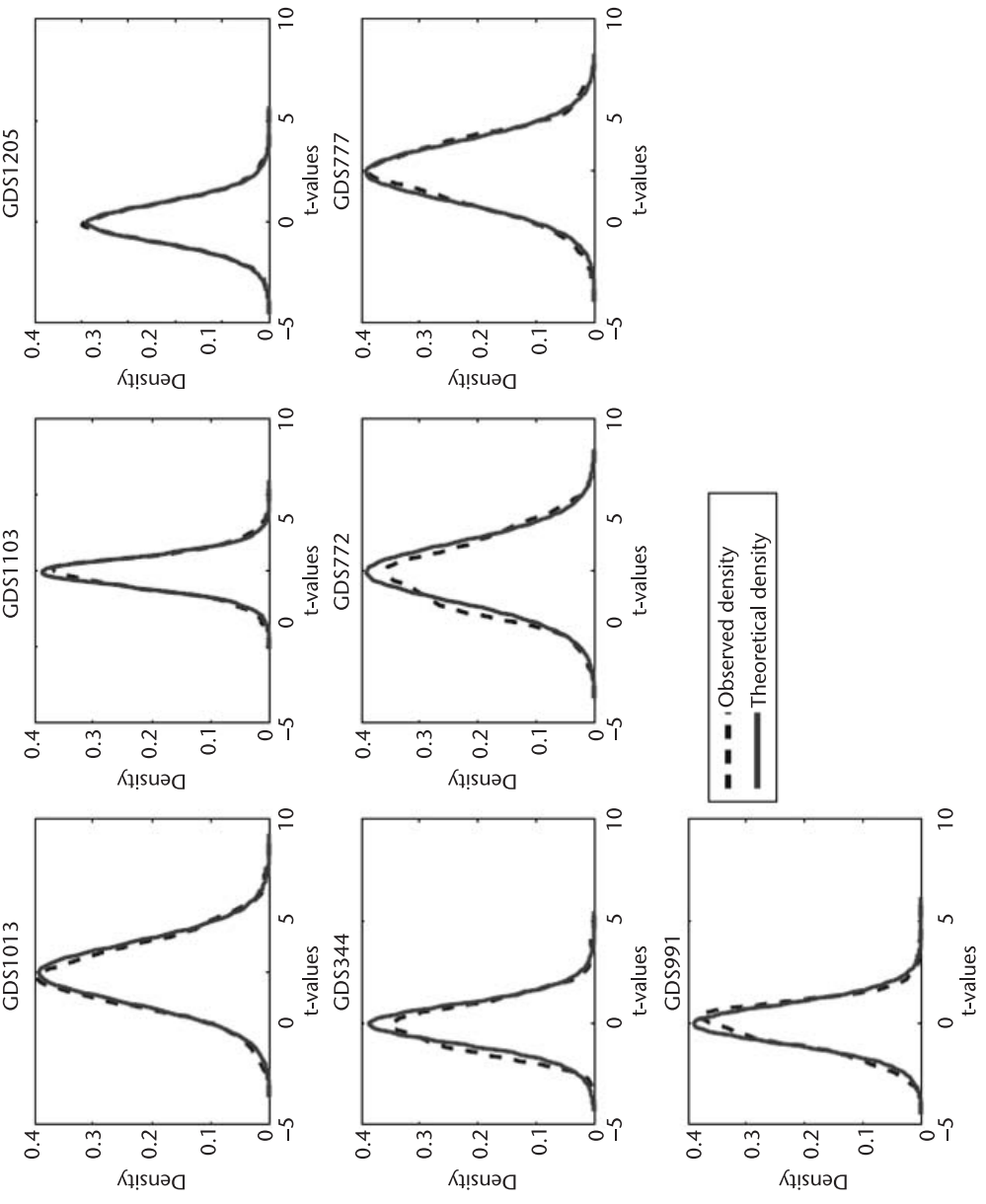


Figure 5.4 Kernel density of observed $\hat{t}$ statistics (dashed line) along with theoretical density (solid line) in seven sample yeast microarray datasets.

5.4.4 Functional-Linkage Network

As illustrated in Figure 5.5, the high-throughput data can be coded into a graph of a functional-linkage n etwork, $G = <V, E>$, where the vertices V of the graph are connected through the edges E. Each vertex represents a protein. The weights of the edges reflect the functional similarities between pairs of the connected proteins. Let $P1$ be the a priori probability of two proteins sharing the same function from microarray data, and $P2$ be the a priori probability from protein-protein interaction data. Then the edge weight is calculated using the negative logarithmic value of the combined probability for the two proteins sharing the same function at the GO-index level of interest

$$\text{Weight of edge} = -\log\left[1-(1-P1)(1-P2)\right] \tag{5.8}$$

As a special case of functional-linage network, a coexpression-linkage network is built only by the coexpression of gene-expression profiles. For a single dataset, we rank all the gene pairs using the p-value defined in (5.5) and choose a fixed number of gene pairs from the top to produce the coexpression-linkage network. For multiple datasets, we rank all gene pairs based on the number of individual p-values that are significant at level 0.01 across multiple datasets [16]; for gene pairs that have the same number of significant p-values, they are ranked by the corresponding metachi-square statistics defined in (5.6). Here, we use metachi-square instead of meta p-value, since the meta p-value for many gene pairs is very close to zero and hard to distinguish computationally (metachi-square, instead of meta p-value, should result in the same order when the degrees of freedom for each gene pair is the same). Then a number of gene pairs are selected from the top to establish the coexpression-linkage network.

The number of gene pairs used to obtain the coexpression-linkage network can be decided in many ways. For instance, the user might simply use the top 200 gene pairs for function prediction, or the Bonferroni correction can be used to obtain a threshold for the individual p-value for single datasets. Also, the magnitude of the Pearson correlation could be considered, combined with the individual p-value, as

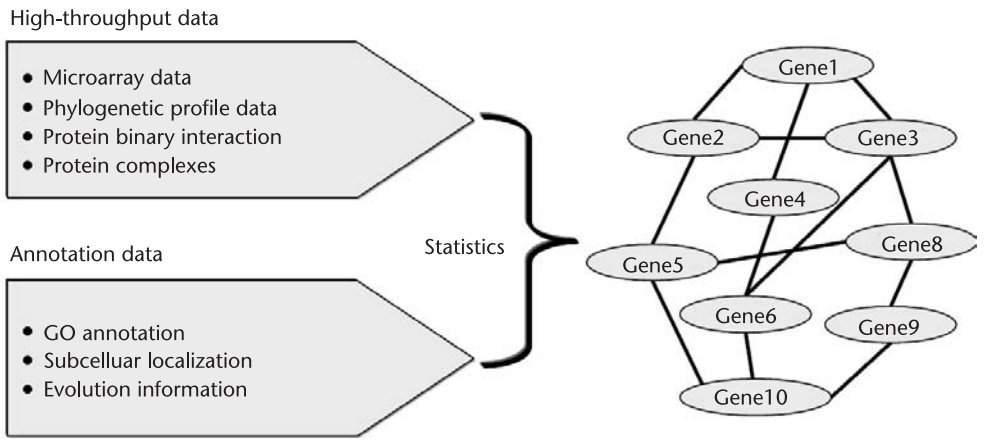


Figure 5.5 Coding high-throughput biological data into a functional-linkage network.

proposed by Lee et al. [16]. For the multiple datasets, the number of significant individual p-values follows a binomial distribution. In other words, p ~binomial (n, 0.01), if a gene pair is not correlated in any of the n datasets. Hence, this binomial distribution can be used to obtain a cutoff value for the number of significant individual p-values. This threshold, together with the cutoff value for the meta p-value using the Bonferroni correction, can be used to choose the gene pairs for the linkage network.

5.5 Function-Prediction Algorithms

Based on the relationship between the functional similarity and statistics of high-throughput data discussed in Sections 5.3 and 5.4, we can predict gene function using various data. The statistical neighbors for each query gene can be obtained from the functional-linkage network, and the union of all functions from the annotated neighbors is assigned to the query gene, each with a likelihood score [14]. Two types of algorithms could be used, based on the network topology and data availability; these are local prediction and global prediction.

5.5.1 Local Prediction

In the local prediction of a gene using its immediate neighbors in the network graph, we follow the idea of guilt-by-association. In other words, if an interaction partner of the studied gene X has a known function, X may share the same function with a probability underlying the high-throughput data between X and its partner. We identify the possible interacting genes for X in each high-throughput data type: protein binary interaction, protein complex interaction (pairwise interaction between any two proteins in a complex), and coexpression profiles with a certain threshold. We assign functions to the unannotated genes on the basis of common functions identified among the annotated interaction partners, using the probabilities described in Section 5.4.3. A gene can belong to one or more functional classes, depending on its interaction partners and their functions. For example, in Figure 5.6, gene X interacts with genes A, B, and C. Assuming F_i, $i = 1, 2, \ldots, n$, represents a collection of all the functions that A, B, and C have, a likelihood score function for X to have function F_i, $G(F_i|X)$ is defined as

$$G(Fi|X) = 1 - \left(1 - P'(S_l|M)\right)^* \left(1 - P'(S_l|B)\right)^* \left(1 - P'(S_l|C)\right) \tag{5.9}$$

where S_l represents the event that two genes have the same function F_i, with GO-index sharing l levels, $l = 1, 2, \ldots, 12$. Given F_i, $P'(S_l|M)$, $P'(S_l|B)$, and $P'(S_l|C)$ are calculated based on probabilities of interaction pairs having the same function for gene-expression correlation coefficient ≥ 0.7 (M), protein binary interaction (B), and protein complex interaction (C), respectively. In each type of high-throughput data, one query gene might have multiple interaction partners with function F_i. Suppose that there are n_M, n_B, and n_C interaction partners, with function F_i in the three types of high-throughput data, respectively. $P'(S_l|M)$, $P'(S_l|B)$, and $P'(S_l|C)$ in (5.9) are calculated as

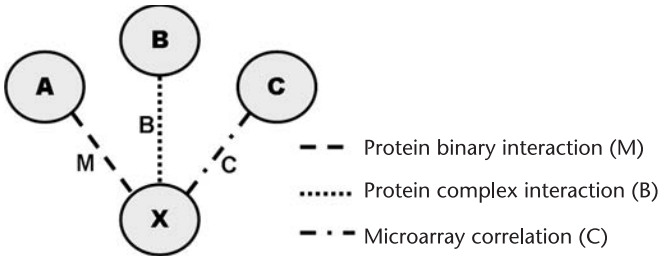


Figure 5.6 Illustration of prediction method. A query gene *X* has interactions with genes *A, B,* and *C,* with known functions. The interactions could be correlation in gene expression (*M*), protein binary interaction (*B*), and protein complex interaction (*C*).

$$P'(S_l|M) = 1 - \prod\left[1 - P_j(S_l|M)\right], \quad j = 1,2,\cdots,n_M \tag{5.10}$$

$$P'(S_l|B) = 1 - \prod\left[1 - P_j(S_l|B)\right], \quad j = 1,2,\cdots,n_B \tag{5.11}$$

$$P'(S_l|C) = 1 - \prod\left[1 - P_j(S_l|C)\right], \quad j = 1,2,\cdots,n_C \tag{5.12}$$

$P_j(S_l|M)$, $P_j(S_l|B)$, and $P_j(S_l|C)$ are estimated probabilities retrieved from the probability curves defined in Section 5.4.3.

We also defined the likelihood score as the reliability score for each function, F_i:

$$\text{Reliability Score} = 1 - \left(1 - P'(S_l|M)\right)^* \left(1 - P'(S_l|B)\right)^* \left(1 - P'(S_l|C)\right) \tag{5.13}$$

The final predictions are sorted based on the reliability score for each predicted GO index. The reliability score represents the probability that the query gene has function F_i, assuming all the evidence from the high-throughput data is independent and only applicable to immediate neighbors in the network.

Gene-function relationship is also highly correlated with sequence similarity, which provides another basis for local prediction using sequence neighbors. With complete sequencing of many genomes and effective ways of finding sequence similarity, making gene-function predictions based on sequence information is becoming more and more reliable.

To apply the sequence information, we select a set of genes *t* whose sequences are similar to the query gene *i*. We apply the hypergeometric model that has the probability density function (pdf) to the gene population with GO-*j* as in (5.14).

$$p = \frac{\binom{M}{x}\binom{N-M}{n-x}}{\binom{N}{n}} \tag{5.14}$$

where N represents the population of all the genes, M represents the number of genes in N that have GO-j, n represents the size of t, and x represents the number of genes in t that have GO-j. Afterward, we calculate the p-value of the hypergeometric distribution. The p-value is the probability that the observed value is greater than a specific variable, reflecting the concentration of the genes in t that have GO-j. The smaller the p-value, the higher the concentration of genes that have GO-j, and hence, the greater the confidence that gene i has the function GO-j. More specifically, the p-value can be calculated using (5.15).

$$\begin{aligned} p-value &= 1-\sum_{j=0}^{x-1}\frac{\binom{M}{j}\binom{N-M}{n-j}}{\binom{N}{n}} = 1-\sum_{j=0}^{x-1}\left[\exp\left(\log\left(\frac{\binom{M}{j}\binom{N-M}{n-j}}{\binom{N}{n}}\right)\right)\right] \\ &= 1-\sum_{j=0}^{x-1}\left[\exp\left(\log\binom{M}{j}+\log\binom{N-M}{n-j}-\log\binom{N}{n}\right)\right] \\ &= 1-\sum_{j=0}^{x-1}\left[\exp\left(\sum_{i=j+1}^{M}\log(i)-\sum_{i=1}^{M-j}\log(i)+\sum_{i=n-j+1}^{N-M}\log(i)-\sum_{i=1}^{N-M-n+j}\log(i)\right.\right. \\ &\qquad \left.\left.-\sum_{i=n+1}^{N}\log(i)+\sum_{i=1}^{N-n}\log(i)\right)\right] \end{aligned} \tag{5.15}$$

where

$$\log\binom{M}{j} = \log(M!)-\log(j!)-\log\left[(M-j)!\right] = \sum_{i=j+1}^{M}\log(i)-\sum_{i=1}^{M-j}\log(i) \tag{5.16}$$

When the neighboring genes have different similarities with the query gene, their contributions to the prediction should be different. In other words, we add a weight for each neighbor.

In Figure 5.7, suppose the similarity for a query gene with neighboring genes, $N1$ to $N8$ is 10, 25, 12, 8, 5, 20, 8, and 12, respectively. Assuming $N2$ and $N6$ share the same GO. Then their contributions to this GO for the query gene are

$$8\times\frac{25+20}{10+25+12+8+5+20+8+12} = \frac{8*45}{100} = 3.6 \text{ instead of } 2.$$

5.5.2 Global Prediction Using a Boltzmann Machine

The major limitation of the local-prediction method is that it uses the information of only immediate neighbors in a network to predict gene function. In some cases, the uncharacterized genes may not have any interacting partner with a known function annotation, and their function cannot be predicted based on the local-prediction method. Therefore, the global properties of the graph are underutilized, since this analysis does not include the links among genes of unknown functions. The functional annotation of uncharacterized genes should not only be decided

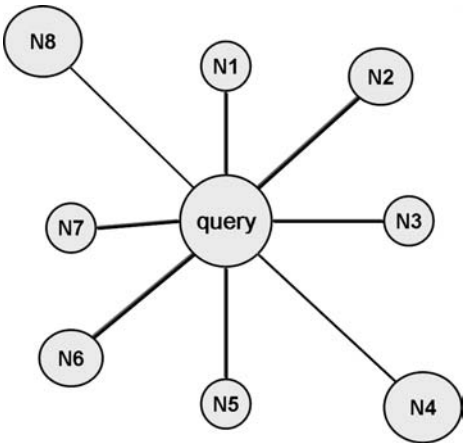


Figure 5.7 A graph of a query gene and its neighbors based on sequence similarities.

by their direct neighbors, but also controlled by the global configuration of the interaction network. Based on such a global-optimization strategy, we developed a new approach for predicting gene function. We used the Boltzmann machine to characterize the global stochastic behaviors of the network. A gene can be assigned to multiple functional classes, each with a certain probability.

We consider a physical system $PS = \{\alpha\}$ (α is a state of the system, each having an energy H_α). In the thermal equilibrium, given a temperature T, each of the possible states α occurs with probability

$$P_\alpha = \frac{1}{Z} e^{-H_\alpha / K_B T} \tag{5.17}$$

where the normalizing factor $Z = \sum_\alpha e^{-H_\alpha / K_B T}$, and K_B is Boltzmann's constant. This is called the Boltzmann-Gibbs distribution. It is usually derived from very general assumptions about microscopic dynamics. In an undirected graphical model with binary-valued nodes, each node (gene) i in the network has only one state value Z_t (1 or 0). For the state at time t, node i (Z_t ,i) has probability $P(Z_t , i = 1 | Z_{t-1}, j \neq i)$ and is given as a sigmoid function of the inputs from all the other nodes at time $t - 1$

$$u_i = \frac{1}{1 + e^{-\beta \sum_{j \neq i} W_{ij} Z_{t-1}}} \tag{5.18}$$

where β is a parameter corresponding to the annealing temperature, and W_{ij} is the weight of the edge connecting genes i and j in the interaction graph. W_{ij} is calculated according to (5.19), by combining the evidence from the gene-expression correlation coefficient ≥ 0.7 (M), protein binary interaction (B), and protein complex interaction (C)

$$W_{ij} = \sum_{F_k} CG\left(F_k | i, j\right) = \sum_{F_k} C\left(1 - \left(1 - P\left(S_l | M\right)\right)\left(1 - P\left(S_l | B\right)\right)\left(1 - P\left(S_l | C\right)\right)\right) \tag{5.19}$$

where S_k represents the event that two genes i and j have the same function F_k(k = 1, 2, …, n), whose GO index has l levels, l = 1, 2, …,12. $P(S_l|M)$, $P(S_l|B)$, and $P(S_l|C)$ were estimated probabilities retrieved from the probability curves calculated in Section 5.4.3. C is the modifying weight

$$C = \begin{cases} 1 & \text{If } j \in \text{ annotated proteins} \\ G\left(F_k \middle| i, j, t' - 1\right) & \text{otherwise} \end{cases} \tag{5.20}$$

To achieve global optimization, we conducted a simulated annealing technique as the following process (Figure 5.8). We set the initial state of all hypothetical genes (nodes) randomly to be 0 or 1, the state of any annotated gene is always 1. Starting with a high temperature, we picked a node i and computed its value u_i,

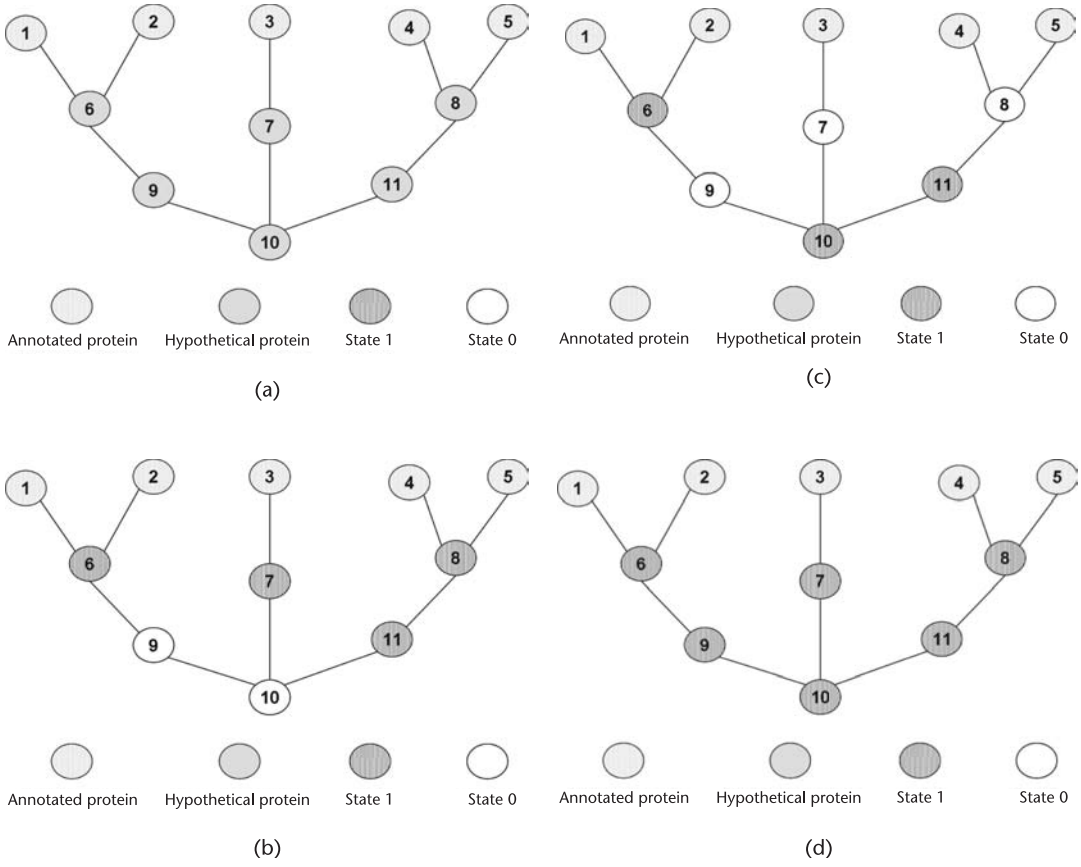


Figure 5.8 Illustration of a global method of function prediction through a simulated annealing technique. (a) A given interaction network where genes 1 to 5 have a known function and genes 6 to 11 are unannotated. (b) In the initial state, the states of all query genes (nodes) are randomly selected to be 0 or 1, and the state of any annotated gene is always 1. State 1 means that the gene is annotated, and state 0 means that the gene's function is unknown. (c) Starting with a high temperature, for each node i, we compute its value u_i, then update its state. Thus, genes 6, 7, 8, and11 can be assigned functions. (d) With the temperature cooling, we again calculate the value u_i of each node i and update its state. All unannotated genes finally can be assigned functions.

then updated its state, until all the nodes in the network reached the equilibrium. With gradual cooling, the system might resettle in a global optimization of the network configuration, if the sum of weights associated with the query genes reaches the maximum value.

5.6 Gene Function-Prediction Experiments

5.6.1 Data Processing

This step includes data collection and preprocessing. Once the required data is collected from online resources, it needs to be processed in order to be used appropriately, as our statistical methods, like many others, are sensitive to data quality. Any poor-quality data might lead to significant false positives in analysis and prediction. For example, in the case of microarray data, normalization and noise removal are important; otherwise, the estimation of the Pearson correlation coefficient (the most commonly used statistic for microarray data) can misrepresent the true correlation between gene-expression profiles.

5.6.2 Sequence-Based Prediction

We did multiple simulations with the hypergeometric model introduced in Section 5.5.1. In each simulation, we randomly selected 100 genes from 5,117 genes with known functions and 1,000 GO annotations from 2,859 total GO annotations. The simulation applies a different p-value threshold to define the neighbors. We compared the gene-function prediction performance between nonweighted p-values and weighted p-values. We sort the p-values in descending order and calculate the ratio between TP (true-positive) and (TP + FP (false-positive)) versus the p-value threshold, as shown in Figure 5.9. It shows that a strict p-value threshold can help enhance the prediction. When the p-value threshold increases from e-6 to e-3, prediction accuracy drops down quickly. In addition, the weighted method is consistently better than the nonweighted one.

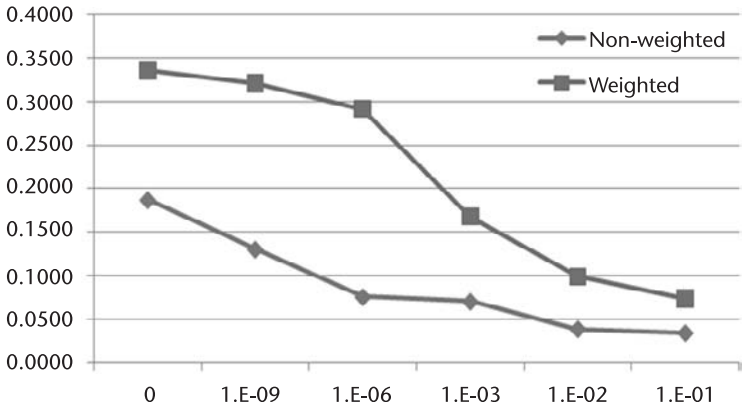


Figure 5.9 Prediction accuracy versus p-value threshold. The *x*-axis is the p-value threshold, and the *y*-axis is TP/(TP+FP).

5.6.3 Meta-Analysis of Yeast Microarray Data

We did a pilot study using 7 independent yeast microarray datasets from the GPL90 platform, including 116 experimental conditions, in total, for all the genes in yeast (Table 5.2). We used the microarray data of 5,419 genes from the GPL90 platform, among which 4,519 genes have GO annotations, whereas the yeast genome GO annotation data was downloaded from the NCBI Gene Expression Omnibus (GEO) Web site, http://www.ncbi.nlm.nih.gov/geo/ [36–38].

Table 5.2 shows the dataset ID, the number of conditions or time points, and the overall experimental condition.

We plotted the conditional probability of the GO functional similarity given an individual p-value (on the log scale) for a single dataset or given the meta p-value for multiple datasets, as shown in Figure 5.10. Although the curves did not differ substantially between a single dataset and the multiple datasets combined, the curve for the meta p-value is much smoother than the curve for any single data, reflecting better statistics with a much larger sample size in the meta-analysis. We also found that there were many more statistically significant pairs using the same threshold for the meta p-values of multiple datasets than those for any single dataset. This suggests that combining multiple datasets using the meta-analysis leads to more discerning power in establishing statistical neighbors for query genes and hence, increases the sensitivity for function prediction.

To confirm this, we applied our function-prediction method to ~10% (500) randomly selected query genes from the yeast genome, using either single datasets or multiple datasets. We compared the sensitivity-specificity plot for 1 dataset and the one using all 7 datasets from Table 5.2. For this purpose, we selected the top 200 neighbors for each query gene to generate the coexpression-linkage network, using either 1 dataset or 7 datasets. We predicted functions for each query gene, one at a time, and then evaluated the sensitivities and specificities of the predictions of all query genes using the sensitivity-specificity curve. For each prediction scheme that corresponds to a particular functional-linkage network and a specific cutoff value for the likelihood scores, the sensitivity and specificity are calculated according to the following definition. We consider assigning a function to a gene as a

Table 5.2 Selection of Microarray Datasets for the Yeast Study

	Dataset	*Columns*	*Experimental Condition*
1	GDS 777	24	Nutrient limitation under aerobic and anaerobic condition effect on gene expression (growth protocol variation)
2	GDS 772	18	Histone deacetylase RPD3 deletion and histone mutation effect on gene regulation (genotype/variation)
3	GDS 344	11	Chitin synthesis (protocol variation)
4	GDS 1205	12	Ssl1 mutant for a subunit of TFIIH response to methyl methanesulfonate (genotype/variation)
5	GDS 1103	12	Leu3 mutant expression profiles (genotype/variation)
6	GDS 991	15	Phosphomannose isomerase PMI40 deletion strain response to excess mannose (dose variation)
7	GDS1013	24	IFH1 overexpression (time course)

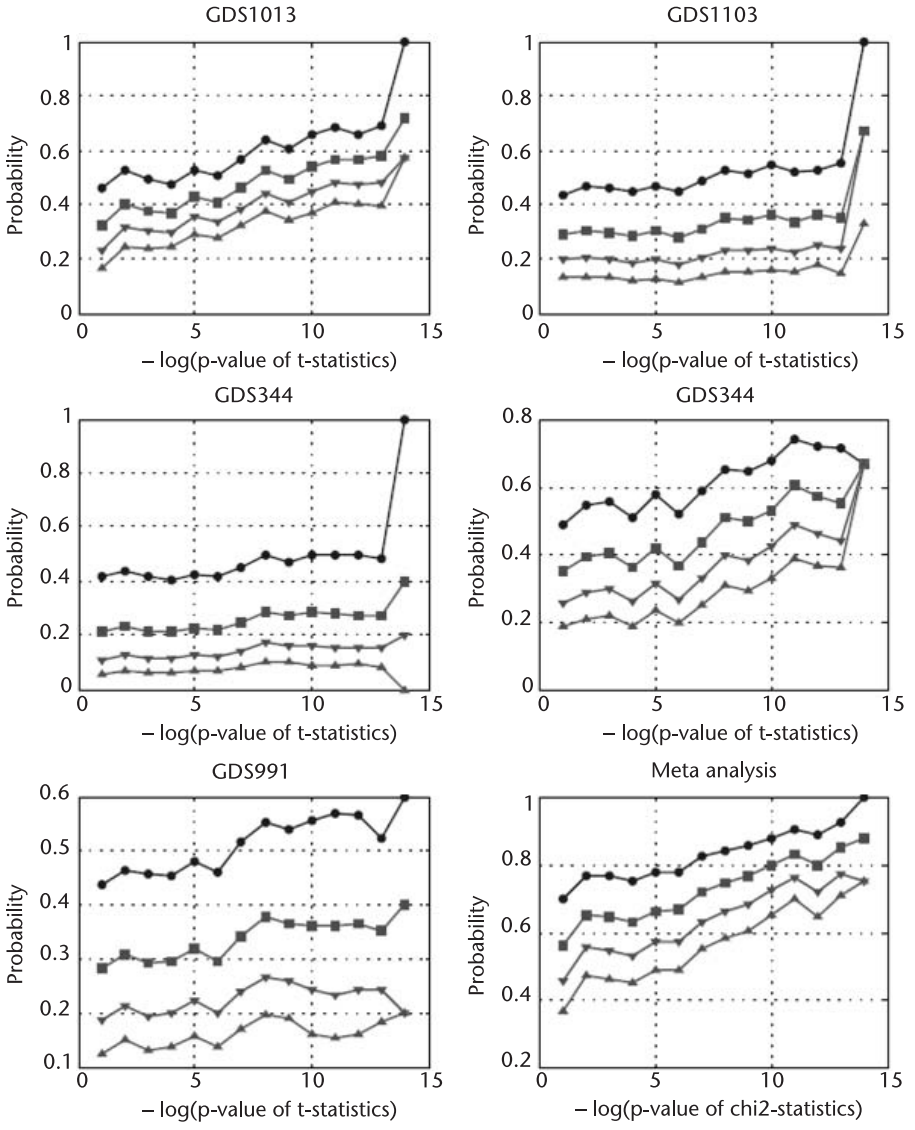


Figure 5.10 Conditional probability of functional similarity given an individual p-value (on log scale) for a single dataset (from five datasets) and given the meta p-value for multiple datasets from yeast.

decision/prediction, which can be verified from the annotation data. There are two types of errors we can make: (1) we assign an incorrect function to a gene, which is a type I error, or a false positive; and (2) we do not assign a known function to a gene, which is a type II error, or false negative. On the other hand, if we assign a correct function to a gene, it is a true positive; if a gene does not have a function and we do not assign it, it is a true negative. We consider all query genes and all available GO IDs in the annotation data and summarize the results in the format of Table 5.3.

Table 5.3 Decision Table for Function Prediction

	Prediction: GO ID Not Assigned	*Prediction: GO ID Assigned*
Known: GO ID not assigned	True negative (TN)	False positive (FP)
Known: GO ID assigned	False negative (FN)	True positive (TP)

By changing the number of predictions selected for each query gene based on the likelihood scores for a fixed coexpression-linkage network, we can obtain a sensitivity-specificity plot, where

$$\begin{aligned} \text{Sensitivity} &= \frac{\sum_{i=1}^{K} TP_i}{\sum_{i=1}^{K} TP_i + \sum_{i=1}^{K} FN_i} \\ \text{Specificity} &= \frac{\sum_{i=1}^{K} TN_i}{\sum_{i=1}^{K} FP_i + \sum_{i=1}^{K} TN_i} \end{aligned} \tag{5.21}$$

where K is the number of query genes, TP_i is the number of correctly predicted functions for gene i, FN_i is the number of known functions that are not predicted for gene i, FP_i is the number of incorrectly assigned functions for gene i, and TN_i is the number of functions among all available GO IDs that are neither known nor predicted for gene i.

We applied our method to the yeast data. Figure 5.11 shows that the meta-analysis using all 7 datasets significantly improved the prediction accuracy over any 1 dataset (4 were chosen as examples). The result suggests that the proposed method of combining multiple microarray datasets using meta-analysis works well.

5.6.4 Case Study: Sin1 and PCBP2 Interactions

When *SIN1* (*MAPKAP1*) was used as the bait in a two-hybrid screen of a human bone marrow cDNA library, its most frequent partner was poly(rC) binding protein 2 (*PCBP2*/*hnRNP-E2*). *PCBP2* associates with the *N*-terminal domain of *SIN1* and the cytoplasmic domain of the *IFN* receptor *IFNAR2*. *SIN1*, but not *PCBP2*, also associates with the receptors that bind *TNF*. *PCBP2* is known to bind to pyrimidine-rich repeats within the 3′ UTR of mRNAs and has been implicated in the control of RNA stability and translation and selective capindependent transcription. RNAi silencing of either *SIN1* or *PCBP2* renders cells sensitive to basal and stress-induced apoptosis. Stress in the form of *TNF* and H_2O_2 treatments rapidly raises the cell content of *SIN1* and *PCBP2*, an effect reversible by inhibiting *MAPK14*.

Human microarray data from the NCBI Gene Expression Omnibus (GEO, www.ncbi.nlm. nih.gov/geo/) SOFT (Simple Omnibus in Text Format) were analyzed to determine the datasets in which *SIN1* and *PCBP2* showed a significant

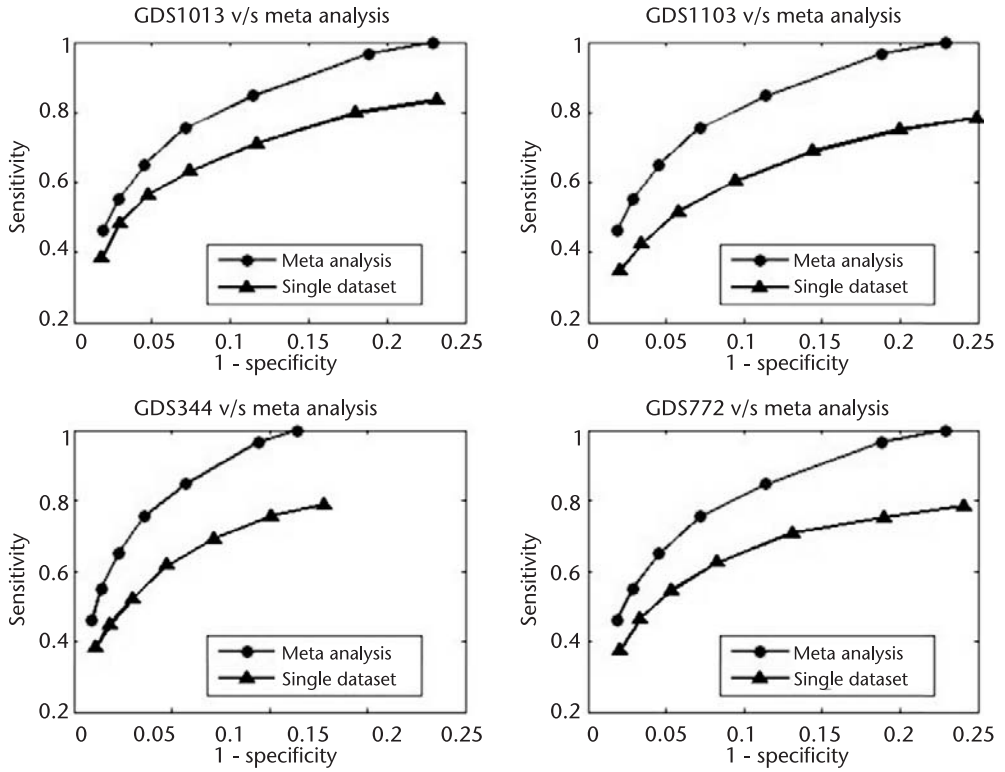


Figure 5.11 Performance comparison between single datasets versus meta-analysis in yeast. In each plot, various cutoff values for the likelihood scores of the prediction functions for the query genes are used to generate different points in the sensitivity-specificity curve. In particular, the 7 points correspond to using the top 50, 100, 200, 400, 800, 1,600, and 3200 predictions for each query gene.

(up or down) change in expression level. Then, the meta-analysis [39] was performed on these datasets to determine which genes were coexpressed with *SIN1*. The analysis created a statistical neighboring linkage network based on functional similarity score and its significance level [17]. Close neighbors (i.e., genes that are coexpressed with *SIN1* over time or in response to treatments) were assumed to have related functions of *SIN1*. Here, the meta-analysis was confined to 1 dataset microarray platform, GPL*96* (i.e., an Affymetrix Gene-Chip Human Genome U133 Array Set HG-U133A) and used 13 curated microarray datasets, each of which had between 50 and 154 arrays. The data was preprocessed and analyzed to provide 2 separate neighbor lists for *SIN1* and *PCBP2*, respectively. The genes in common to each list with a significance level of $P < 0.01$ were then identified and ranked, based on associated confidence scores. The annotations of these identified genes are shown in Figure 5.12.

The meta-analysis of human microarray data supports the hypothesis that *SIN1* plays a central, directive role in controlling apoptosis [40]. With few exceptions, genes and pathways regulated in concert with *SIN1* are involved in reacting

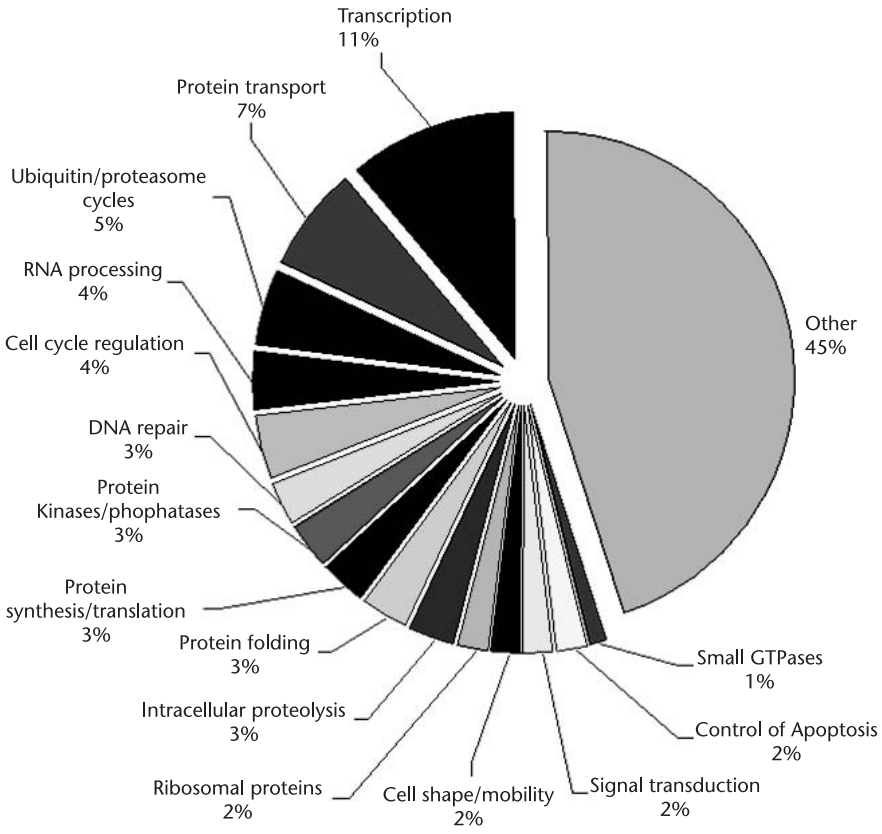


Figure 5.12 Classes of annotated genes that demonstrate expression profiles similar to both *SIN1* and *PCBP2*. GO biological processes were used to identify functional classes of the 984 annotated coexpressed genes.

to various forms of stress. *SIN1* appears to occupy an important node in a network of pathways that safeguard cells against environmental affronts and subsequently allow the cells either to die or to recover from damage. *PCBP2*, which is as vital as *SIN1* in shielding against apoptosis, is also expressed coordinately with genes that encode large numbers of cell-survival, as well as cell-death, factors.

5.7 Transcription Network Feature Analysis

Gene ontology can also be used for GO-enrichment analysis to identify various network features in different networks, such as a relevance network, an associate network, or a regulatory network [8, 41–48]. Here, we take the example of a regulatory network to show the application of ontology to the analysis of regulons. A *regulon* is a set of genes that are regulated by the same transcription factor. The function of any regulon on a subnetwork can be summarized by finding significant enriched GO terms. We conducted GO enrichment within the Arabidopsis network, reconstructed using a meta-analysis of microarray data.

5.7.1 Time Delay in Transcriptional Regulation

Having a successful application in constructing a functional-linkage network, we applied meta-analysis for studying the regulatory relationship between a transcription factor and its targets. It has been shown that the activation of a regulator under stress conditions usually occurs earlier than the activation of its targets [49, 50]. A noticeable time difference exists among changes in concentrations of the regulator mRNA, the regulator protein, and the mRNAs of its targets. Therefore, in order to infer a regulatory relationship from the microarray data, we develop a chemical kinetic model to theoretically fit the time lag between these events (Figure 5.13) [51].

5.7.2 Kinetic Model for Time Series Microarray

The regulator-protein concentration can be modeled by the following chemical kinetic equation, without considering posttranslational regulation:

$$\frac{dR_p}{dt} = K_{tran}R_m - K_pR_p \tag{5.22}$$

where R_p is the regulator-protein concentration, R_m is the regulator-mRNA concentration, K_{tran} is the apparent rate of mRNA translation, and K_p is the turnover rate of the regulator protein. Accordingly, the time course of the target mRNA concentration can be modeled as

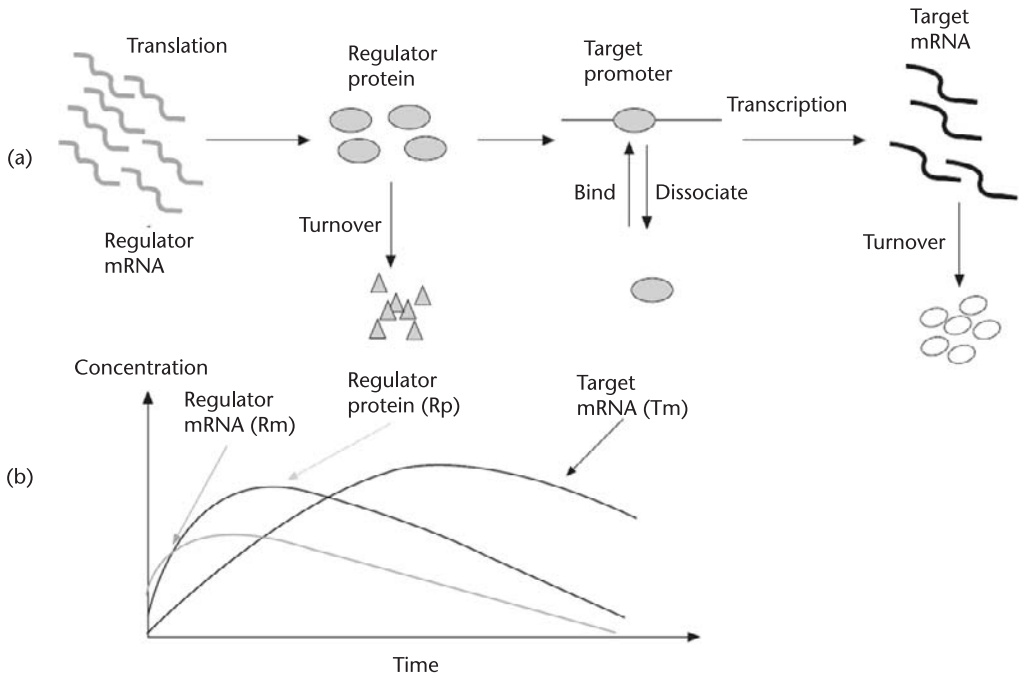


Figure 5.13 Schematics of the transcriptional regulation process. (a) Steps of chemical reactions considered in the kinetic model and (b) schematics of the temporal curves of the regulator protein and target mRNA in response to regulator mRNA changes.

$$\frac{dT_m}{dt} = B_t + f\left(R_p\right) - K_t T_m \tag{5.23}$$

where T_m is the concentration of the target mRNA, B_t is the basal transcription rate of the target gene, K_t is the turnover rate of the target mRNA, and $f(R_p)$ measures the regulated transcription rate. For simplicity, $f(R_p)$ takes the following form:

$$f\left(R_p\right) = K_{act} R_p \tag{5.24}$$

Usually, what is reported in transcription-profiling experiments is not the absolute concentration of mRNA, but rather a fold change, compared to the basal transcription level of that gene. Thus, we define relative changes of R_m and T_m as $R_{m'}$ and $T_{m'}$,

$$R_{m'} = \frac{R_m}{R_{mbasal} - 1} \tag{5.25}$$

$$T_{m'} = \frac{T_m}{T_{mbasal} - 1} \tag{5.26}$$

where T_{mbasal} and R_{mbasal} are the basal concentrations of the regulator protein and target mRNA, respectively. Combining the above equations leads to the following second-order ordinary differential equation:

$$\frac{d^2\left(T_{m'}\right)R_p}{dt^2} + \left(K_t + K_p\right)\frac{d\left(T_{m'}\right)}{dt} + K_t K_p T_{m'} = \gamma R_{m'} \tag{5.27}$$

where

$$\gamma = \frac{K_{act} K_{tran} R_{mbasal}}{T_{mbasal}} \tag{5.28}$$

To predict the target of a specific regulator, we can solve (5.27) to obtain the theoretical target-behavior curve, and then find the genes with mRNA levels similar to those of the theoretical curve, which will be identified as the potential targets of that regulator.

5.7.3 Regulatory Network Reconstruction

The kinetic model for the time-lag problem, along with the meta-analysis technique to combine inferences from different microarray datasets, provided basic elements for constructing gene regulatory subnetworks around transcription factors. We evaluated our method on an Arabidopsis gene expression dataset containing 497 arrays measuring responses to various stress conditions [19, 50, 52] and compared with the online available database AgrisDB [53, 54]. In this experiment, wild-type Arabidopsis plants were subjected to stress treatments for various periods (1, 2, 5, 10, and 24 hours), and extracted mRNA samples were hybridized to a cDNA microarray. For meta-analysis, we used 9 separate tissue-specific microarray datasets, as gene expression is typically tissue-specific. That is, each tissue typically has its own set of genes expressed, although there are overlaps among tissues. Tissuewise partitioning of microarray data and combining it using meta-analysis shows ~7

Table 5.4 Regulatory Network Construction for Arabidopsis, Using Two Different Techniques

Method	*Network Size*	*Confirmed Pairs*
Regression	~ 40,000	16
Meta-analysis	~ 12,000	35

times improvement in the network over the one from using the causal-regression model [19], as shown in Table 5.4. This indicates that consistent relationships between a transcription factor and a target, across most tissues, indicate a more robust prediction for gene regulation.

In Table 5.4, the first column shows the method used to build the network, the second column shows the network size (number of edges in the network), and the last column shows the confirmed edges from the Agris database.

5.7.4 GO-Enrichment Analysis

Using this global network, we predicted ~179 genes that are significantly regulated by the E2F transcription factor in at least 7 out of 9 tissues, as mentioned in Section 5.7.3, and we identified new candidate genes. This transcription factor provides essential activities for coordinating the control of cellular proliferation and cell fate.

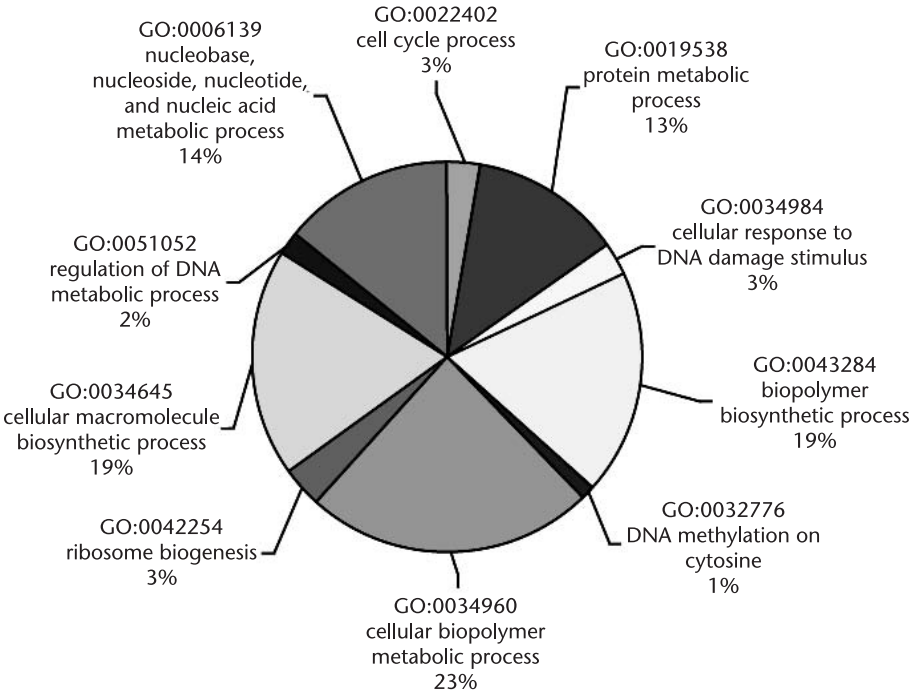


Figure 5.14 Distribution of putative genes regulated by the E2F transcriptions factor in Arabidopsis. The description of these GO terms shows that the major categories of these processes include cell cycle, DNA replication, and DNA repair.

The significantly enriched functional annotation terms across E2F targeted genes show that E2F plays a crucial role in the control of cell-cycle progression and regulates the expression of genes required for the G1/S transition. These include genes encoding DNA replication proteins, enzymes involved in nucleotide synthesis, and components of the origin-recognition complex, as shown in Figure 5.14.

5.8 Software Implementation

5.8.1 GENEFAS

We developed a GENE Function Annotation System (GENEFAS), which is computational software with a graphical user interface for gene-function prediction by integrating information from protein-protein interactions, protein complexes, microarray gene-expression profiles, and annotations of known proteins. GENEFAS can provide biologists a workspace, for their organisms of interest, to integrate different types of experimental data and annotation information. It is freely available for download at http://digbio.missouri.edu/genefas. GENEFAS allows a user to generate hypotheses and predict functions for their genes of interest and to get a global view of the relationship among genes. Users can retrieve information based on a search of an open reading frame (ORF) name, gene name, or annotation keyword. The software also facilitates sequence-based searches and provides users with the option to select different data types and upload both public and private datasets for integration in function prediction. Users can get a global understanding of the relationships among different gene products by viewing the *neighboring genes*, defined to be neighbors on the basis of the distance calculated from high-throughput data and functional-classification pie charts. GENEFAS provides biologists with testing and training capabilities based on different datasets and evaluates the performance based on sensitivity and specificity plots.

5.8.2 Tools for Meta-Analysis

Function-prediction tools using meta-analysis of microarray data are available from http://digbio.missouri.edu/meta_analyses/. All programs were written using ANSI C language, and they are compatible with both Linux, as well as Windows, operating systems.

5.9 Conclusion

This chapter introduced various aspects of GO and its applications in gene function and regulatory-network characterization. GO provides a controlled vocabulary to map functions of genes into identifiers in any organism. This notation makes the computational method feasible to manipulate gene functions in terms of ontology or certain types of mapping. GO tremendously saved the time for other researchers to collect up-to-date function annotation from the literature, as it is continuously updated, and new versions are made available on a monthly basis. There are also

some other types of ontologies, such as the KEGG ontology. KEGG (Kyoto Encyclopedia of Genes and Genomes) is a collection of online databases dealing with genomes, enzymatic pathways, and biological chemicals. More ontologies are introduced in Chapter 1 of this book.

GO offers the most comprehensive sets of relationships to describe gene/protein activities. However, GO also has some limitations. For example, some GO terms are generic and not informative for biological studies, although GO has been improved with more specific function details over the years. Furthermore, GO's choice to segregate gene ontology to subdomains of molecular function, biological process, and cellular component creates some limitations [55]. With further developments of gene ontology to overcome these limitations, new computational methods for gene-function prediction will also emerge.

Acknowledgements

We would like to thank our collaborators, Drs. R. Michael Roberts and Jeffery Becker. We would also like to thank Yu Chen for his early involvement in this work. This study was supported by USDA/CSREES-2004-25604-14708 and NSF/ITR-IIS-0407204 and a Monsanto internship for Gyan Srivastava.

References

[1] Troyanskaya, O. G., "Putting Microarrays in a Context: Integrated Analysis of Diverse Biological Data," *Brief Bioinform,* Vol. 6, No. 1, 2005, pp. 34–43.

[2] Watson, J. D., R. A. Laskowski, and J. M. Thornton, "Predicting Protein Function from Sequence and Structural Data," *Curr Opin Struct Biol*, Vol. 15, No. 3, 2005 pp. 275–284.

[3] Barutcuoglu, Z., R. E. Schapire, and O. G. Troyanskaya, "Hierarchical Multi-Label Prediction of Gene Function," *Bioinformatics*, Vol. 22, No. 7, 2006, pp. 830–836.

[4] Deng, M., T. Chen, and F. Sun, "An Integrated Probabilistic Model for Functional Prediction of Proteins," *J Comput Biol*, Vol. 11, Nos. 2–3, 2004, pp. 463–475.

[5] Lanckriet, G. R., et al., "Kernel-Based Data Fusion and Its Application to Protein Function Prediction in Yeast," *Pac Symp Biocomput*, Honolulu, January, 6–10, 2004, pp. 300–311.

[6] Marcotte, E. M., et al., "A Combined Algorithm for Genome-Wide Prediction of Protein Function," *Nature*, Vol. 402, No. 6757, 1999 pp. 83–86.

[7] Pavlidis, P., et al., "Learning Gene Functional Classifications from Multiple Data Types," *J Comput Biol*, Vol. 9, No. 2, 2002, pp. 401–411.

[8] Brazhnik, P., A. de la Fuente, and P. Mendes, "Gene Networks: How to Put the Function in Genomics," *Trends Biotechnol*, Vol. 20, No. 11, 2002, pp. 467–472.

[9] Ashburner, M., et al., "Gene Ontology: Tool for the Unification of Biology. The Gene Ontology Consortium," *Nat Genet*, Vol. 25, No. 1, 2000, pp. 25–29.

[10] Chen, Y., and D. Xu, "Global Protein Function Annotation Through Mining Genome-Scale Data in Yeast Saccharomyces Cerevisiae," *Nucleic Acids Res*, Vol. 32, No. 21, 2004, pp. 6414–6424.

[11] Chen, Y., and D. Xu, "Understanding Protein Dispensability Through Machine-Learning Analysis of High-Throughput Data," *Bioinformatics*, Vol. 21, No. 5, 2005, pp. 575–581.

[12] Choi, J. K., et al., "Combining Multiple Microarray Studies and Modeling Interstudy Variation," *Bioinformatics*, Vol. 19, Supplement 1, 2003, pp. i84–i90.

[13] Hughes, T. R., et al., "Functional Discovery via a Compendium of Expression Profiles," *Cell*, Vol. 102, No. 1, 2000, pp. 109–126.

[14] Joshi, T., et al., "GeneFAS: A Tool for Prediction of Gene Function Using Multiple Sources of Data," *Methods Mol Biol*, Vol. 439, 2008, pp. 369–386.

[15] Kishino, H., and P. J. Waddell, "Correspondence Analysis of Genes and Tissue Types and Finding Genetic Links from Microarray Data," *Genome Inform*, Genome Inform Ser Workshop, Vol. 11, 2000, pp. 83–95.

[16] Lee, H. K., et al., "Coexpression Analysis of Human Genes Across Many Microarray Data Sets," *Genome Res*, Vol. 14, No. 6, 2004, pp. 1085–1094.

[17] Rhodes, D. R., et al., "Meta-Analysis of Microarrays: Interstudy Validation of Gene Expression Profiles Reveals Pathway Dysregulation in Prostate Cancer," *Cancer Res*, Vol. 62, No. 15, 2002, pp. 4427–4433.

[18] Rhodes, D. R., et al., "Large-Scale Meta-Analysis of Cancer Microarray Data Identifies Common Transcriptional Profiles of Neoplastic Transformation and Progression," *Proc Natl Acad Sci USA*, Vol. 101, No. 25, 2004, pp. 9309–9314.

[19] Seki, M., et al., "Functional Annotation of a Full-Length Arabidopsis cDNA Collection," *Science*, Vol. 296, No. 5565, 2002, pp. 141–145.

[20] Zhou, X. J., et al., "Functional Annotation and Network Reconstruction Through Cross-Platform Integration of Microarray Data," *Nat Biotechnol*, Vol. 23, No. 2, 2005, pp. 238–243.

[21] Joshi, T., and Xu, D., "Quantitative Assessment of Relationship Between Sequence Similarity and Function Similarity," *BMC Genomics*, Vol. 8, No. 1, 2007, p. 222.

[22] Altschul, S. F., et al., "Gapped BLAST and PSI-BLAST: A New Generation of Protein Database Search Programs," *Nucleic Acids Res*, Vol 25, No. 17, 1997, pp. 3389–3402.

[23] O'Brien, K. P., M. Remm, and E. L. Sonnhammer, "Inparanoid: A Comprehensive Database of Eukaryotic Orthologs," *Nucleic Acids Res*, Vol. 33, Database Issue, 2005, pp. D476–D480.

[24] Pellegrini, M., et al., "Assigning Protein Functions by Comparative Genome Analysis: Protein Phylogenetic Profiles," *Proc Natl Acad Sci USA*, Vol. 96, No. 8, 1999, pp. 4285–4288.

[25] Finn, R. D., et al., "Pfam: Clans, Web Tools and Services," *Nucleic Acids Res*, Vol 34, Database Issue, 2006, pp. D247–D251.

[26] Mulder, N. J., et al., "InterPro, Progress and Status in 2005," *Nucleic Acids Res*, Vol. 33, Database Issue, 2005, pp. D201–D205.

[27] Glazko, G., A. Gordon, and A. Mushegian, "The Choice of Optimal Distance Measure in Genome-Wide Datasets," *Bioinformatics*, Vol 21, Supplement 3, 2005, pp. iii3–iii11.

[28] Warnat, P., R. Eils, and B. Brors, "Cross-Platform Analysis of Cancer Microarray Data Improves Gene Expression Based Classification of Phenotypes," *BMC Bioinformatics*, Vol. 6, 2005, p. 265.

[29] Stevens, J. R., and R.W. Doerge, "Combining Affymetrix Microarray Results," *BMC Bioinformatics*, Vol. 6, 2005, p. 57.

[30] Reverter, A., et al., "Joint Analysis of Multiple cDNA Microarray Studies via Multivariate Mixed Models Applied to Genetic Improvement of Beef Cattle," *J Anim Sci*, Vol. 82, No. 12, 2004, pp. 3430–3439.

[31] Magwene, P. M., and J. Kim, "Estimating Genomic Coexpression Networks Using First-Order Conditional Independence," *Genome Biol*, Vol. 5, No. 12, 2004, p. R100.

[32] Culhane, A. C., et al., "MADE4: An R Package for Multivariate Analysis of Gene Expression Data," *Bioinformatics*, Vol. 21, No. 11, 2005, pp. 2789–2790.

[33] Eisen, M. B., et al., "Cluster Analysis and Display of Genome-Wide Expression Patterns," *Proc Natl Acad Sci USA*, Vol. 95, No. 25, 1998, pp. 14863–14868.

[34] Kim, S. K., et al., "A Gene Expression Map for Caenorhabditis Elegans," *Science*, Vol 293, No. 5537, 2001, pp. 2087–2092.

[35] Segal, E., et al., "Module Networks: Identifying Regulatory Modules and Their Condition-Specific Regulators from Gene Expression Data," *Nat Genet*, Vol. 34, No. 2, 2003, pp. 166–176.

[36] Barrett, T., et al., "NCBI GEO: Mining Millions of Expression Profiles—Database and Tools," *Nucleic Acids Res*,Vol. 33, Database Issue, 2005, pp. D562–D566.

[37] Barrett, T., et al., "NCBI GEO: Mining Tens of Millions of Expression Profiles—Database and Tools Update," *Nucleic Acids Res*, Vol. 35, Database Issue, 2007, pp. D760–D765.

[38] Barrett, T., and R. Edgar, "Mining Microarray Data at NCBI's Gene Expression Omnibus (GEO)," *Methods Mol Biol*, Vol. 338, 2006, pp. 175–190.

[39] Park, T., et al., "Combining Multiple Microarrays in the Presence of Controlling Variables," *Bioinformatics*, Vol. 22, No. 14, 2006, pp. 1682–1689.

[40] Ghosh, D., et al., "A Link Between SIN1 (MAPKAP1) and Ply(rC) Binding Protein 2 (PCBP2) in Counteracting Environmental Stress," *Proc Natl Acad Sci USA*, Vol. 105, No. 33, 2008, pp. 11673–11678.

[41] Stuart, J. M., et al., "A Gene-Coexpression Network for Global Discovery of Conserved Genetic Modules," *Science*, Vol. 302, No. 5643, 2003, pp. 249–255.

[42] Barabasi, A. L., and Z. N. Oltvai, "Network Biology: Understanding the Cell's Functional Organization," *Nat Rev Genet*, Vol. 5, No. 2, 2004, pp. 101–113.

[43] de la Fuente, A., P. Brazhnik, and P. Mendes, "Linking the Genes: Inferring Quantitative Gene Networks from Microarray Data," *Trends Genet*, Vol. 18, No. 9, 2002. pp. 395–398.

[44] Gachon, C. M., et al., "Transcriptional Co-Regulation of Secondary Metabolism Enzymes in Arabidopsis: Functional and Evolutionary Implications," *Plant Mol Biol*, Vol 58, No. 2, 2005, pp. 229–245.

[45] Ideker, T., et al., "Discovering Regulatory and Signalling Circuits in Molecular Interaction Networks," *Bioinformatics*, Vol. 18, Supplement 1, 2002 pp. S233–S240.

[46] Toh, H., and K. Horimoto, "Inference of a Genetic Network by a Combined Approach of Cluster Analysis and Graphical Gaussian Modeling," *Bioinformatics*, Vol. 18, No. 2, 2002, pp. 287–297.

[47] Opgen-Rhein, R., and K. Strimmer, "From Correlation to Causation Networks: A Simple Approximate Learning Algorithm and Its Application to High-Dimensional Plant Gene Expression Data," *BMC Syst Biol*, Vol. 1, 2007 p. 37.

[48] Lee, T. I., et al., "Transcriptional Regulatory Networks in Saccharomyces Cerevisiae," *Science*, Vol. 298, No. 5594, 2002, pp. 799–804.

[49] Haake, V., et al., "Transcription Factor CBF4 Is a Regulator of Drought Adaptation in Arabidopsis," *Plant Physiol*, Vol. 130, No. 2, 2002, pp. 639–648.

[50] Seki, M., et al., "Monitoring the Expression Pattern of 1300 Arabidopsis Genes Under Drought and Cold Stresses by Using a Full-Length cDNA Microarray," *Plant Cell*, Vol. 13, No. 1, 2001, pp. 61–72.

[51] Yugi, K., et al., "A Microarray Data-Based Semi-Kinetic Method for Predicting Quantitative Dynamics of Genetic Networks," *BMC Bioinformatics*, Vol. 6, 2005, p. 299.

[52] Schmid, M., et al., "A Gene Expression Map of Arabidopsis Thaliana Development," *Nat Genet*, Vol. 37, No. 5, 2005, pp. 501–506.

[53] Palaniswamy, S. K., et al., "AGRIS and AtRegNet. A Platform to Link Cis-Regulatory Elements and Transcription Factors into Regulatory Networks," *Plant Physiol*, Vol. 140, No. 3, 2006, pp. 818–829.

[54] Davuluri, R.V., et al., "AGRIS: Arabidopsis Gene Regulatory Information Server, an Information Resource of Arabidopsis Cis-Regulatory Elements and Transcription Factors," *BMC Bioinformatics*, Vol. 4, 2003, p. 25.

[55] Pal, D., "On Gene Ontology and Function Annotation," *Bioinformation*, Vol. 1, No. 3, 2006, pp. 97–98.

CHAPTER 6

Mapping Genes to Biological Pathways Using Ontological Fuzzy Rule Systems

Mihail Popescu and Dong Xu

In this chapter, we provide another example of application in which ontologies play a transformational role in the underlying algorithms. We show how the ontological similarity described in Chapter 2 is employed in a fuzzy rule system, resulting in a new type of rule system, called an ontological fuzzy rule system (OFRS). After we define the OFRS, we illustrate its application with a bioinformatics example related to mapping genes to gene networks.

6.1 Rule-Based Representation in Biomedical Applications

Rule-based knowledge systems have been popular in the medical informatics community, due to their transparency and their relative ease of development for the expert physician. A typical example is MYCIN, an expert system developed in the 1970s to diagnose and treat infections in humans [4]. An example of a MYCIN rule is given below:

IF
  the *gram* of the organism is "gram negative" AND
  the *morphology* of the organism is "rod" AND
  the *aerobicity* is "anaerobic"
THEN
  the organism is "Bacteroides" with certainty 0.6

The MYCIN expert system operates in a forward-chaining manner. The rules are fired by *syntactically* matching the values of the input variables ("gram negative," "rod," and "anaerobic," in the above example) to the variables (*gram*, *morphology,* and *aerobicity*) from the antecedent of the rules. A certainty factor that reflects the physician's confidence in the rule is associated with each rule's output. However, the expert-system rules do not fire for inputs that are semantically related

to the value of the variable specified in the rule (for example, if the value of the *aerobicity* input were "aerointolerant"). The semantic imprecision inherent in human communication can be externally controlled if the expert system is administered by a physician. However, an autonomous agent that acts independently in a Semantic Web environment will have difficulty in dealing with semantic imprecision.

To answer the terminological imprecision in the Semantic Web context, various domain ontologies have been constructed (http://www.obofoundy.org). Many description logic rule based engines, such as KAON2, FACT++, or RACER, which use rules described by ontological terms, are under development [2]. Their scalability and ability to handle complex problems are continuously improving; however, their capacity to handle imprecision is limited, since they, too, use a syntactic matching of ontology terms.

The imprecision of the system inputs was addressed in a simple fashion in engineering by the use of fuzzy rule systems. The variables in the rule antecedents (usually referred to as linguistic variables) are fuzzy sets represented by membership functions. For example, the variable *stature* may have values, such as "short," "average," and "tall," that can be represented by the membership functions shown in Figure 6.1.

The variable from the rule consequence may be a fuzzy set (Mamdani fuzzy rule systems) or a function (Takagi-Sugeno-Kwang fuzzy rule systems). A typical example of a fuzzy rule may be:

IF
  a person's *stature* is "tall"
THEN
  the person's *speed* is "fast"

A fuzzy rule such as the one shown above is tolerant to input imprecision: different statures such as 1.75 and 1.80 will not significantly (abruptly) change the output of the rule. Fuzzy rules are performing well in the area of control systems engineering, in which the inputs are few (typically less than 5) and numeric [6]. The greatest limitation of the fuzzy rule systems is that they do not handle symbolic information.

Due to its data-intensive nature, bioinformatics has initially favored a data-driven approach to knowledge representation. Large amounts of data stored in an

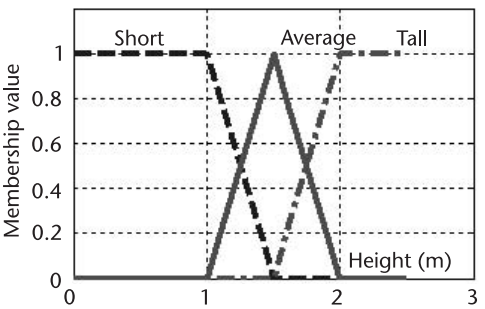


Figure 6.1 Memberships for three values of the variable *stature*: "short," "average," and "tall."

ever-increasing number of databases have been used to solve biological problems, with a variety of computational intelligence techniques such as data mining, fuzzy logic, evolutionary computing, and neural networks. In this case, knowledge was typically extracted from the database in the form of association rules (see, for example, Chapter 7). The data in bioinformatics often has several characteristics that make a rule-based representation challenging: large volume, mixed symbolic and numeric variables, and imprecision intrinsic to biological phenomena and laboratory techniques [18]. Hence, a bioinformatics rule-based system has to handle both numeric and symbolic imprecision and be scalable.

To deal with the above challenges, we proposed [8, 9] the concept of an ontological fuzzy rule system (OFRS). An ontological fuzzy rule system can have two types of input: numeric (numbers) and symbolic (words). While the numeric inputs require that their related linguistic variables be represented as membership functions, the symbolic ones require only the definition of a natural similarity for the construction of a membership function. The main idea of OFRS is to view the ontological similarity between concepts (as discussed in Chapter 2) as a fuzzy membership function, which we will discuss next.

6.2 Ontological Similarity as a Fuzzy Membership

The idea of using word similarities for defining fuzzy sets was advanced by Zadeh [20] in his "computing with words" paradigm. There, Zadeh stated that most human concepts have a granular structure in which the knowledge granules are arranged in taxonomies. Zadeh's *granule* is a generic object (or set of objects) that may refer to any human perception and knowledge, such as time, space, or, in our case, biology. Here, we view the ontology concepts as knowledge granules that may contain, for example, several different syntactic variants (strings) of the concept. Our granules (genes, pathways, and so on) are contained in taxonomies specific to biology, such as GO and KEGG. Then, Zadeh mentions that a granule is "a fuzzy set drawn together by similarity." The similarity between concepts can be computed, if the granule taxonomy is known, using similarity measures such as those described in Chapter 2. Our concept of a granule is depicted in Figure 6.2.

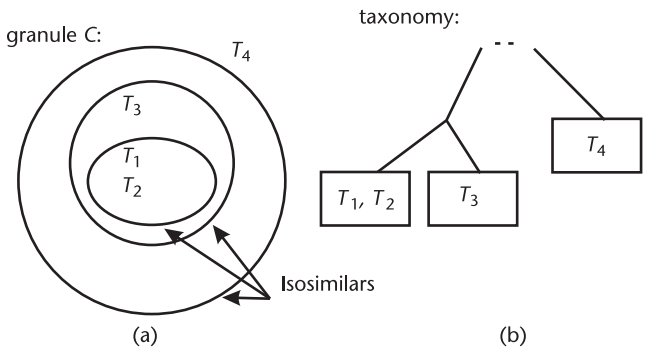


Figure 6.2 A granule that represents a concept *C* (a) contains several terms arranged in a taxonomy (b).

Assume that the two terms, T_1 and T_2, at the center of the granule in Figure 6.2, precisely describe the concept C. That is, their concept memberships are $\mu_{1,2}(C) = 1$. They may be seen as different string variants (synonyms) that describe the concept C. In this case, the semantic similarity between them is 1. On the other hand, the similarity between T_1 and T_3, a more distant term in the taxonomy, can be computed using the formulas shown in Chapter 2. Since T_4 is even more remote in the taxonomy from T_1 and T_2 than T_3, it is expected that its membership in C be smaller, that is $\mu_4(C) < \mu_3(C)$. Consequently, the circles around T_1 and T_2 can be viewed as isosimilars, the geometric loci of all terms equally similar to T_1 and T_2.

Andreasen [1] (also see Chapter 8) further refined the interpretation of similarity between two ontology terms as a fuzzy membership by observing that, while the similarity measure in an ontology is usually symmetrical (that is, $s(T_1,T_2)=s(T_2,T_1)$), the related fuzzy memberships might not be. The reason resides in the difference between generalization ("tyrosine kinases are kynases") and specialization ("kinases are tyrosine kinases"). While the first statement is true ("high" membership value of term "tyrosine kinases" in "kinases"), the latter is only partially true ("medium" membership value of "kinases" in "tyrosine kinases").

To answer the above observation, we have to employ a nonsymmetrical similarity measure for computing term similarity. In this chapter, we use a simple path-based similarity inspired from [1]. The similarity between two terms, T_1 and T_2, is computed as

$$s_{12}(T_1, T_2) = \max_{\{P_i\}} \prod_{j \in P_i} w_{ij} \tag{6.1}$$

where $\{P_i\}$ is the set of all possible paths connecting T_1 and T_2 in an ontology, and w_{ij} is the weight assigned to the arc j from path P_i (see Figure 6.3).

We note that s_{12} from (6.1) is not a similarity relation, because, in general, it is not symmetrical (e.g., in Figure 6.3, $s(T_1,T_3) \neq s(T_3,T_1)$). We consider only two types of weights here: specialization weights (downward from the ancestor node to the descendent node) with a value of 0.4 and generalization weights (upward from the descendent to the ancestor) with a value of 0.9. The interpretation of the

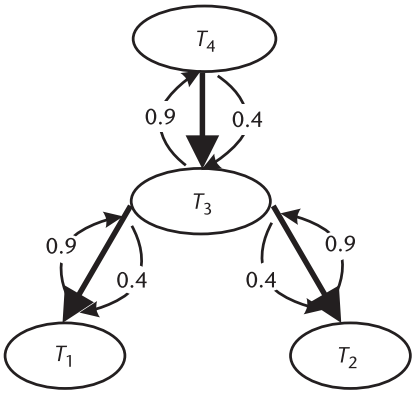


Figure 6.3 Example of a path-based computation of the similarity (fuzzy membership) between two ontology terms. The thin arcs represent the weight-assignment process, while the thick arcs represent the ontological relation *is a* (*has_a* relations are ignored).

weights is that T_1 is T_3 to the extent of 0.9, but T_3 is T_2 only to the extent of 0.4. The weight values (0.4 and 0.9) were chosen arbitrarily (as in [1]), but they can be learned from a corpus based on term co-occurrences [17].

Example 6.1 Consider the ontology snippet from Figure 6.3 with two terms, T_1 and T_2, that have a common parent T_3. There are two ways of getting from T_1 to T_2, (T_1,T_3,T_2) and (T_1,T_3,T_4,T_3,T_2), hence the similarity between T_1 and T_2 is according to (1) s_{12}=max{w($T_1 \rightarrow T_3$) × w($T_3 \rightarrow T_2$), w($T_1 \rightarrow T_3$) × w($T_3 \rightarrow T_4$) × w($T_4 \rightarrow T_3$) × w($T_3 \rightarrow T_2$)} = max{0.9 × 0.4, 0.9 × 0.9 × 0.4 × 0.4}=0.36.

A more general case is when C_1 is an object that has some properties described by a set of terms (such as T_1 and T_2). The question is then to compute the similarity, s(C_1,C_2), of the object C_1 to another object C_2, described by terms from the same ontology. The resulting similarity is interpreted as the object C_2 is a C_1 to the extent of $s(C_1,C_2)$. One method for computing the similarity between two objects C_1={T_{11},..., T_{1n}} and C_2={T_{21}, ...,T_{2m}}, described by ontology terms, is the normalized average approach:

$$s(C_1,C_2) = \frac{s_a(C_1,C_2)}{\max\{s_a(C_1,C_1), s_a(C_2,C_2)\}}, \qquad s_a(C_1,C_2) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} s(T_{1i},T_{2j})}{mn} \tag{6.2}$$

where $s(t_{1i}, T_{2j})$ is computed using (6.1). The normalization of the average ensures that $s(C_1,C_1)$=1. Other possible similarity measures are described in Chapter 2.

Example 6.2 Let us compute the similarity between two objects, C_1={T_1, T_2} and C_2={T_3, T_4}. First, observe that if we do not consider the ontological relations between the terms T_i, $i \in$ {1,2,3,4}, the similarity between the two objects is 0. Assume that, using some term-similarity measure, such as Formula (1) in some ontology, we have: $s(T_1,T_2)$ = 0.2, $s(T_1,T_3)$ = 0.3, $s(T_1,T_4)$ = 0.4, $s(T_2,T_3)$ = 0.5, s(T_2,T_4) = 0.6, s(T_3,T_4) = 0.7, and $s(T_i,T_i)$ = 1 for any $i \in$ {1,2,3,4}. Then, using (6.2), we obtain $s_a(C_1,C_2)$ = 0.45, $s_a(C_1,C_1)$ = 0.8, $s_a(C_2,C_2)$ = 0.85 and, finally, $s(C_1,C_2)$=0.45/max{0.8, 0.85} = 0.53.

An interesting method for defining the distance between entities in first-order logic (FOL) was found in [3]. The method was based on comparing the predicates used to describe two objects, and it was used in the conceptual clustering system, KBG.

6.3 Ontological Fuzzy Rule System (OFRS)

We define an ontological fuzzy rule system (OFRS) by analogy to a Mamdani fuzzy rule system (FRS). A typical Mamdani FRS with n inputs and 1 output variable has the following form [6]:

Rule 1: **IF** x_1 is G_{11} **AND** ... **AND** x_n is G_{1n} **THEN** y is P_1
Rule m: **IF** x_1 is G_{m1} **AND** ... **AND** x_n is G_{mn} **THEN** y is P_m (6.3)

where G_{ij} and P_i, $i \in [1,m]$, and $j \in [1,n]$, are fuzzy sets, $\{x_i\}$ is the inputs variable, and y is the output variable, respectively. The fuzzy sets G_{ij} are possible values for the variable x_i, while P_i is a possible value for the output y. G_{ij} and P_i are usually represented using membership functions of the type shown in Figure 6.1. A Mamdani FRS for $m = 2$ and $n = 2$ is shown in Figure 6.4.

The computation of the FRS output, $y_0 \in R$, for two inputs $x_{10}, x_{20} \in R$, is performed as follows:

The memberships, $w_{ij}=G_{ij}(x_{i0})$, are computed for each rule i and input j, $i,j \in \{1,2\}$. Second, the activation, a_i, of rule i is computed as $a_i = w_{i1} \oplus w_{i2}$, where $\oplus$ is an AND type operator. In most applications $\oplus$ is *minimum* (that is, $a_i = \min\{w_{i1},w_{i2}\}$), but other choices are possible (see [6]). It is also possible for the variables in the rule antecedent (left-hand side) to be joined by an OR type operator. In this case, the typical operator employed is *maximum*, that is, $a_i = \max\{w_{i1},w_{i2}\}$. After computing the activation, the output of each rule is computed as a_iP_i, which is the shaded area of each output membership P_i shown in Figure 6.4. The fuzzy output of the system, P_{sum}, is computed by aggregating the individual rule outputs as $P_{\text{sum}} = \max\{a_1P_1, a_2P_2\}$. Here, too, more choices of aggregating operators, other than *max*, are possible. The final step of the computation is called *defuzzification,* and it consists of reducing the output fuzzy set P_{sum} to a number. Among the most used defuzzification procedures are the center of gravity (COG, y_0 in Figure 6.4) and mean of maximum (MOM, y_1 in Figure 6.4). Using the COG procedure, the output of the FRS, y_0, is computed as the center of the area under the membership function P_{sum}. By employing the MOM algorithm, the output y_1 is computed as the center of the region where the membership P_{sum} is the maximum.

An OFRS is similar to the FRS described above, except

1. Some/all input fuzzy sets G_{ij} are replaced by ontology terms or by objects described by ontology terms. The variables related to these terms or objects are called *symbolic variables*. As opposed to the numeric variables, for

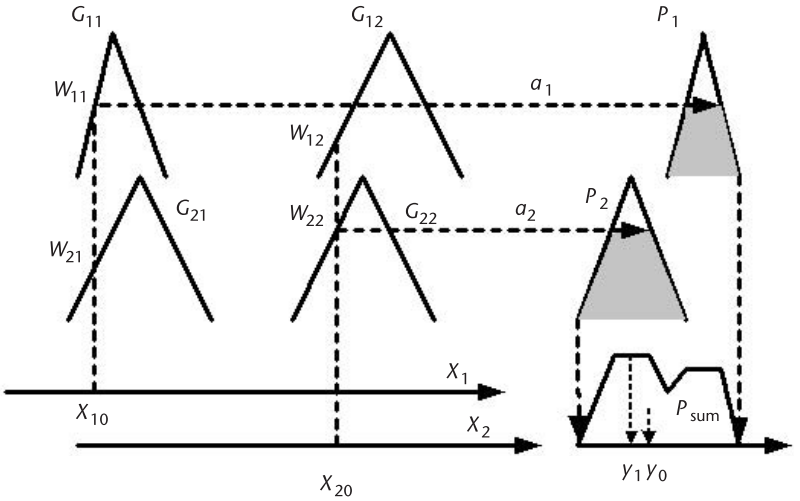


Figure 6.4 A Mamdani fuzzy rule system with $n = 2$ rules and $m = 2$ inputs. The inputs and the output of the system are real numbers.

which the values are fuzzy sets represented by functions, the symbolic variables have values that consist of ontology terms or objects represented by ontology terms. Similarly, the output fuzzy sets P_i are replaced with terms from the same ontology (output ontology) in order to allow the assessment of relatedness (similarity). The output of the OFRS consists of a term or a set of terms from the output ontology.

2. The membership, w_{ij}, of the input x_i in the G_{ij} is now computed, based on the similarity of the two terms (objects), using (6.1) or (6.2) as $w_{ij} = s(x_i, G_{ij})$.
3. The aggregation of the rule output has to take into account that the final output of the OFRS should be a term or a set of terms. Consequently, the defuzzification procedure has to summarize the output of the set of rules in few terms. Similar to the defuzzification procedures described above for FRS, we mention two summarization procedures. The first procedure, somewhat similar to MOM, chooses as output the term P_k, with the maximum activation, where k is given by

$$k = \arg\max_{i=1,m} \{a_i\} \tag{6.4}$$

The wining rule activation a_k may be used as the confidence of the OFRS output.

This first case does not use the term *similarity* as mentioned in item 2. The second procedure, denoted here as ontological COG (OCOG), tries to find the "center"of all output terms P_i. In this case, the output term P_k is chosen as

$$k = \arg_{i=1,m} \max \left\{ \sum_{j=1}^{n} \frac{a_i s\left(P_1, P_j\right)}{n} \right\} \tag{6.5}$$

The OCOG procedure has two desired properties. First, the ancestors of a term contribute to the term's importance, and hence, to its chance of winning in (6.5). Second, since the similarity relation is nonsymmetrical, that is s(ancestor, child) < s(child, ancestor) (as explained in Figure 6.3), the child does not contribute as much to its ancestor. Hence, the OCOG procedure tends to choose the more specific term as the final output. However, the average similarity value in (6.5) has a tendency of producing low membership values. A possible solution for this problem is to use an OWA-type operator [19], such as summing only the first n' most similar terms where $n' < n$.

Example 6.3 Consider the following medical OFRS:

rule 1: **IF** x_1="back aches" **AND** x_2="high fever" **THEN** P_1="spinal meningitis"
rule 2: **IF** x_1="aches" **AND** x_2="moderate fever" **THEN** P_2="flu"

In the above OFRS, x_1 is a symbolic variable (a symptom), and x_2 is a numeric one (the fever in degrees Fahrenheit). Assume, as in Figure 6.3, that s("back aches," "aches") = 0.9, s("aches," "back aches") = 0.4, s("flu," "spinal meningitis") = 0.1, s("spinal meningitis," "flu") = 0.2, and all self-similarities $s(T_i, T_i) = 1$. Moreover,

assume that the memberships for fever ("moderate" and "high") are similar to the ones from Figure 6.1, but adapted to the range of the fever. For the purpose of this example, we assume the following memberships: w("moderate," 100) = 0.9 and w("high,"100) = 0.5.

For an input $\boldsymbol{x}$ = {x_1 = "aches," x_2 = 100}, rule 1 will have an activation a_1 = min{0.9,0.5} = 0.5 and rule 2 will have a_2 = min{1,0.9}=0.9. Using (6.4), the output of the system is given by argmax{0.5, 0.9} = 2, that is, P_2 = "flu," with confidence 0.9. Note, that the activation of the first rule is also significant (0.5).

If we use (6.5) for the same input, we get argmax{(0.5*1+0.5*0.2)/2, (0.9*1+0.9*0.1)/2} = argmax{0.3, 0.49} = 2; hence, the output is P_2, with confidence 0.49. Here we see the tendency of the average operator from (6.5) to produce low confidence values (0.49 is much smaller than 0.9, which was obtained in the first case.

6.4 Application of OFRSs: Mapping Genes to Biological Pathways

The problem of mapping a set of genes to pathways often arises in microarray experiments in which we would like to know which regulatory networks can explain about the observed gene-expression patterns. The majority of gene-mapping applications [5,15,16] employ statistical algorithms based on syntactically matching the gene names in a given pathway. There are two main disadvantages of the name-matching approach. First, since a given gene may have several names, it might not match the (string) variant that appears in the pathway representation. Second, some sequences represented on the microarray may belong to an unknown gene, hence having no available name for matching. This is often the case when a new organism is sequenced. One possible solution to the previous problems is to use the Gene Ontology (GO) annotation of a gene instead of its name in the pathway search process. The GO terms for an annotated gene may be retrieved from http://www.geneontology.org (see Chapter 1). If the sequence has not been annotated yet, we can use an automated gene-function prediction method (see Chapter 5) to compute the appropriate GO terms.

By investigating the biological pathways, the biologists aim at understanding the processes that underlay gene expression and, hence, the cause of an abnormal cell condition (such as cancer). One of the databases that store the current biological-pathway knowledge is KEGG (Kyoto Encyclopedia of Genes and Genomes, http://www.genome.jp/kegg). For a certain organism (*Homo sapiens* or *Arabidopsis thaliana,* in our case), KEGG contains all the related known biological pathways and the genes associated with each of them. In addition, KEGG contains the links between the genes, a fact that we ignore in the applications described in this chapter. By neglecting the links between genes, the KEGG database can be seen as an OFRS, similar to the one shown in (3).

In Sections 6.4.1 and 6.4.2, we will describe two approaches to mapping genes to regulatory pathways using OFRSs. In the first approach [9], we employed an OFRS that has the terms in the antecedent linked by OR operators (disjunctions). In the second approach [8], we used an OFRS that has the terms in the antecedent connected by AND operators (conjunctions).

6.4.1 Mapping Gene to Pathways Using a Disjunctive OFRS

6.4.1.1 Disjunctive OFRS

The disjunctive rules of the OFRS used in [9] have the form

$$\text{Rule } i\text{: IF } x \text{ is } G_{i1} \text{ OR ... OR } x \text{ is } G_{in} \text{ THEN } P_{i1} \quad i \in [1,m] \tag{6.6}$$

The activation, a_i, of each rule for an input x is computed as

$$a_i = \underset{j=1,n}{OR}(w_{ij}) = \min_{j=1,n}\{w_{ij}\} \tag{6.7}$$

where OR is a disjunctive operator, and w_{ij} is calculated using the ontological similarity between x and G_i, $s(x, G_{ij})$. In this application, we do not use any rule aggregation, as in (6.4) and (6.5). Instead, we use the activation of each rule a_i to associate each input gene to an m-dimensional output vector $A = (a_1, ..., a_m)$, $A \in R^m$.

For the case of mapping genes to pathways, the input variable of the OFRS is a gene annotated with terms from the Gene Ontology (GO), and the output variable is a KEGG pathway (http://www.genome.ad.jp/kegg). The concrete form of the above OFRS rule is

$$\textbf{IF } gene \text{ is } \text{GENE}_{i1} \textbf{ OR ... OR } gene \text{ is } \text{GENE}_{in} \textbf{ THEN } pathway \text{ is } \text{PATH}_i \tag{6.8}$$

where GENE_{ij} are genes identified by KEGG as being present in pathway PATH_i. In fact, the OFRS consists in the KEGG pathway database itself. The OFRS has just one gene as input variable. The output of the OFRS is the membership of the input gene in a pathway PATH_i. As mentioned above, the OFRS (6.8) maps each gene into an m-dimensional feature vector that represents the membership in each pathway. Next, we present an example of computing the activation of a rule (6.8) for a given input gene.

Example 6.4 Given the rule "**IF** gene is *BCL2* **OR** gene is *APAF1* **THEN** pathway is *APOPTOSIS*" we compute the rule activation for *CASP9*. Using the GO Web site, http://www.geneontology.org, we obtain the following annotations (only two shown) for each of the three H. sapiens genes mentioned above: *CASP9*={GO:0008632: apoptotic program, GO:0008635: caspase activation via cytochrome c}, *BCL2*={GO:0006916: antiapoptosis, GO:0006959: humoral immune response}, and *APAF1*={GO:0008635: caspase activation via cytochrome c, GO:0042981: regulation of apoptosis}. Using the term-similarity method from (6.1) and the GO snippet from Figure 6.5, we obtain the following term-similarity matrix inTable 6.1. For example, the similarity between GO terms GO:0008635 and GO:0008632 is computed as $0.9^4 \times 0.4 = 0.26$.

The "relatedness" of the *CASP9* to *BCL2*, w_1, is given by their GO similarity computed using the normalized pairwise similarity (6.2): $w_1 = s(CASP9, BCL2) = s(\{GO:0008632, GO:0008635\}, \{GO:0006919, GO:0006959\}) = [(0.11+0.32+0.02+0.001)/4]/ [\max\{(1+1+0.04+0.02)/4, (1+1+0.03+0.26)/4\}] = 0.2$. Similarly, the membership of *CASP9* in *APAF1* is $w_2 = 0.72$. The rule activation (6.7) is $a =$

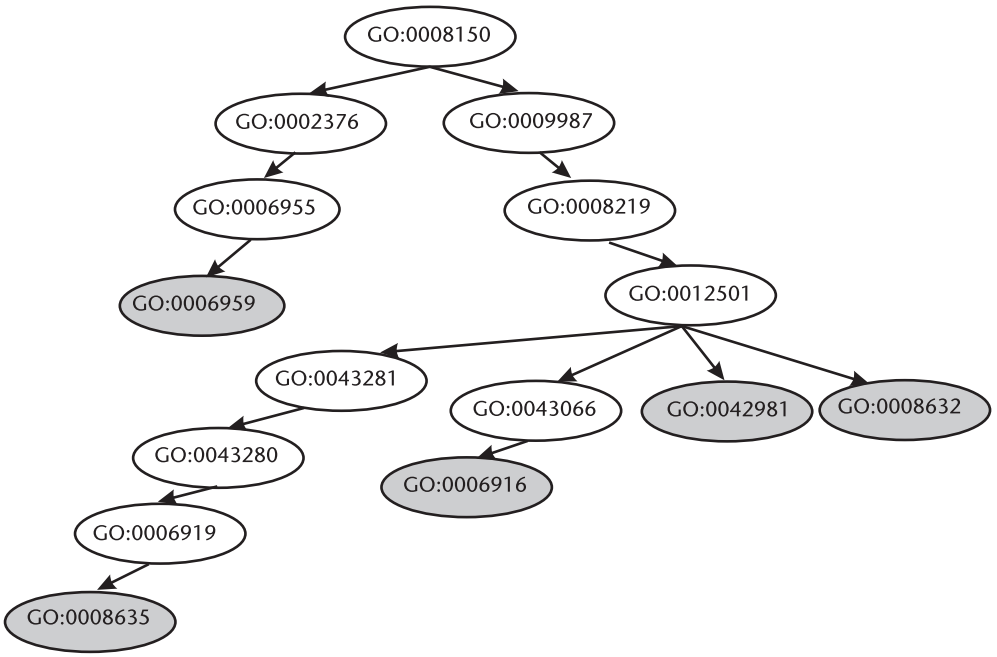


Figure 6.5 Gene Ontology snippet for the terms used in Example 6.4.

Table 6.1 GO Term-Similarity Matrix Computed with (6.1) and the GO Snippet from Figure 6.5

	GO:0008632	*GO:0008635*	*GO:0006916*	*GO:0006959*	*GO:0042981*
GO:0008632	1	0.03	0.15	0.04	0.36
GO:0008635	0.26	1	0.11	0.03	0.26
GO:0006916	0.32	0.11	1	0.04	0.32
GO:0006959	0.02	0.001	0.01	1	0.02
GO:0042981	0.36	0.03	0.15	0.04	1

$\max\{w_1, w_2\} = 0.72$. The rule activation is high, as it should be, since *CASP9* is part of the apoptosis pathway.

6.4.1.2 Gene-Mapping Algorithm

The input of the mapping algorithm is a set of GO annotated genes $Q = \{q_i\}_{i=[1,N]}$. The goal of the algorithm is to find the KEGG pathways (their numbers and identities) that are involved in the expression of genes from the set Q. The pathway-prediction algorithm has the following steps:

1. Compute the activation a_{ij} of each gene q_i, $i \in [1,N]$, in pathway j, $j \in [1,m]$, using (6.7). As a result each gene i is described by a pathway activation (feature) vector $A_i = (a_{i1},\ldots,a_{im}) \in R^m$.
2. Compute the gene-similarity matrix, $S = \{s_{ij}\}_{i,j \in [1,N]}$, as

$$s_{ij} = \frac{A_i^T \cdot A_j^T}{\sqrt{\left|A_i^T\right|\left|A_j^T\right|}} \tag{6.9}$$

where A^T denotes that the vector A was thresholded with a threshold T (that is, if $a_{ij} < T$, then $a_{ij} = 0$). The thresholding operation was performed in order to remove the noise (pathways with residual activation). The best threshold was determined experimentally [9] to be $T = 0.5$.

3. Use a clustering algorithm, together with a cluster validity measure, to assess the most likely number C of pathways (clusters) present in Q. We used the fuzzy C-means algorithm [10] to cluster the genes represented by the feature vectors $\{S_i\}_{i=1,N}$ into C clusters, where $S_i = (s_{i1}, \ldots, s_{iN})$ and the partition coefficient [10] to estimate the number of clusters. We found that it is more reliable to cluster the similarity matrix S using fuzzy C-means, rather than the feature vectors $\{A_i\}$ directly. Another possible approach to clustering a similarity matrix is to use a relational clustering algorithm such as non-Euclidean relational fuzzy C-means [11], together with a relational clustering validity measure, such as the correlation cluster validity [12] (as shown in Chapter 3).
4. *Step 4*. Assume is the set of indices from cluster $c \in [1,C]$, where $\sum_{c=1}^{C} |I_k| = N$ and $|I|$ denotes the cardinality of I. The pathway that is more likely for the genes in cluster c to be active in is the one for which the sum of the activations in cluster c is maximum. If we denote this pathway by P_k, $k \in [1,m]$, then k is obtained using

$$k = \arg\max_{j=1,m} \left\{Sum_j\right\} \tag{6.10}$$

where $Sum_j = \sum_{i \in I_k} a_{ji}$ To produce more than one candidate pathway for a cluster, we can consider the pathway that has the second highest sum activation in the cluster, and so on.

5. The evaluation of the mapping that was performed using the detection rate (DR, sensitivity) is computed as

$$DR = \frac{no\_pathways\_correct\_predicted}{total\_no\_correct\_pathways} \tag{6.11}$$

The false prediction rate (FPR) is computed as

$$FPR = \frac{no\_pathways\_erroneously\_predicted}{total\_no\_pathways\_predicted} \tag{6.12}$$

For example, if the KEGG IDs of the correct pathway are {10, 940, 3050}, and our prediction is {10, 940, 3030, 4070}, then $DR = 0.66$ and $FPR = 0.5$. We note

that, since we ignore that the pathways 3050 and 3030 are strongly related, our *DR* estimate is conservative.

We also estimate the p-value of our *DR* prediction by randomly assigning the membership of the *N* genes in *C* clusters and recomputing the detection rate, DR^*. We perform the random assignment 1,000 times, resulting in a set of 1,000 random detection rates, $\{DR_j^*\}_{j=1,1000}$. Then, the p-value is calculated as

$$p-value = \frac{\left\{no\_of\_DR_j^*, DR_j^* > DR\right\}}{1000} \tag{6.13}$$

that is, the number of the random detection rates higher than our *DR* (obtained by clustering S_i's) divided by 1,000. The p-value is a measure of the reliability of our classifier. If the p-value is low (e.g., lower than 0.05), a low detection rate might be due to a gene set that is hard to predict and not to a bad prediction method.

6.4.1.3 Testing the Mapping Algorithm on 10 *H. Sapiens* Gene Sets

The algorithm described in Section 6.4.1.2 was used with KEGG pathways for Homo sapiens and Arabidopsis thaliana as fuzzy rule system databases. Usually, the fuzzy rules are set up by domain experts. In our case, the memberships of genes in pathways (the rule base) were determined by biologists and stored in the KEGG database. An alternative way of building the OFSR is to employ an item set (association rules) mining method for finding the rules.

For testing, we used the July 2006 version of the KEGG pathway database for *H. sapiens*. We tested the algorithm using 10 sets of 15 genes each, randomly selected (without replacement) from KEGG pathways that have more than 50 genes. The reason for this condition was that we tried to minimize the impact on the whole pathway at the extraction of 5 genes from it. We found 23 such pathways out of the $m = 181$ H. sapiens pathways considered. Each set of genes was extracted from three pathways (5 genes per pathway).

The results obtained on the H. sapiens test set are presented in Table 6.2. The prediction was made by considering one candidate pathway (the one that had the maximum activation sum) per cluster and using a feature threshold of $T = 0.5$.

As we can see from Table 6.2, over-clustering (like in the sets numbered 2, 3, 8, and 10) leads to an increase in false predictions. Sometimes clusters may be merged, if they predict the same pathway. However, we leave pruning strategies for further research.

We mentioned that predicting the right pathways (as for set 6) does not necessarily mean that we assigned all the genes to the correct pathways in the process. For example in set 6, we assigned only 13 out of 15 genes (87%) to the correct pathways. In Figure 6.6 we show the gene-similarity matrix computed using (6.9) and the pathway features for set 6.

We see that genes 4 and 12 (circled) exhibit more similarity to the genes from pathway 2 (gene index 6–10) than to their own pathways (gene index 1–5, and gene index 11–15, respectively). On average, we predicted about 45% of the genes in the right pathway.

Table 6.2 Pathway Prediction Results for 10 *H. Sapiens* Test Gene Sets Using One Candidate Pathway per Cluster

Set #	*# Pathways Predicted, C,(out of 3)*	*DR*	*FPR*	*# Genes in the correct pathway (out of 15)*
1	3	0.67	0.33	9
2	5	0.67	0.60	4
3	5	1.00	0.40	7
4	3	0.67	0.33	10
5	3	0.67	0.33	5
6	3	1.00	0	13
7	3	0.33	0.67	3
8	4	0.33	0.75	2
9	3	0.67	0.33	9
10	4	0.67	0.50	5
Mean		0.66	0.43	6.7

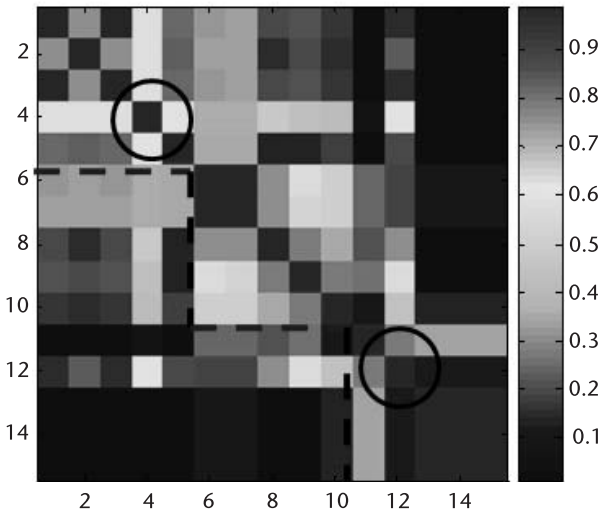


Figure 6.6 Similarity matrix for the 15 genes selected in case 6 from Table 6.2. Genes 4 and 12 (circled) will be erroneously grouped by fuzzy C-means in pathway 2, (indices 6–10), instead of pathways 1 (indices 1–5) and 3 (indices 11–15), respectively.

6.4.1.4 Predicting the Pathways Involved in an *Arabidopsis Thaliana* Microarray Dataset

The pilot dataset used for further testing of our method consisted of 526 *A. thaliana* genes selected in a microarray experiment. In this experiment, we considered m = 115 pathways from the July 2006 KEGG version. Out of 526 genes in the input set, we found only 438 to be annotated using a GO term. Since we did not use any automated annotation software in this work, we removed the 88 unannotated genes

from the experiment. To determine the most probable number of clusters, we used the partition coefficient [10] that resulted in $C = 8$ group of genes. In Table 6.3, we show the KEGG IDs for the three representative pathways found for each of the 8 clusters.

We see that most of the clusters are coherent; that is, the pathway candidates for a cluster are very similar. For example, cluster 1 has 7 genes, and the candidate pathways are oxidative phosphorilation (190), ATP synthesis (193), and photosynthesys (195) (which are obviously related, since 193 is included in 190, and 195 and 190 are both related to the energy metabolism). Similarly, cluster 5 has 25 genes, and the candidates pathways are DNA polymerase (3030), transcription factor (3022), and ribosome (3010), which are all involved in the DNA replication process. Finally, cluster 8 has 69 genes involved in valine, leucine, and isoleucine degradation (280) and biosynthesis (290).

The similarity matrix for the 438 genes is shown in Figure 6.7.

In Figure 6.7, we can distinguish the 8 clusters described in Table 6.3. Furthermore, by inspecting Figure 6.7 more carefully, we observe that the genes (circled) from cluster 4 (around index 200) and from cluster 7 (around index 350) seem to be highly similar. Table 6.3 confirms this observation, since they share the second pathway candidate: sphingolipid metabolism (KEGG ID 600).

Although this method gave encouraging results for our pilot dataset, it has two potential problems that derive from the fact that it maps one gene at a time. First, by mapping one gene at a time, it is not considering the dependencies between the genes in a pathway. Second, mapping one gene at a time results in a low signal-to-noise ratio, due to the noise produced by the similarity to various genes other than itself. Consequently, a better approach would be to map groups of genes at a time. Since it is impossible to know a priori the grouping of the genes, this approach relies on an evolutionary strategy for estimating the number of pathways and their

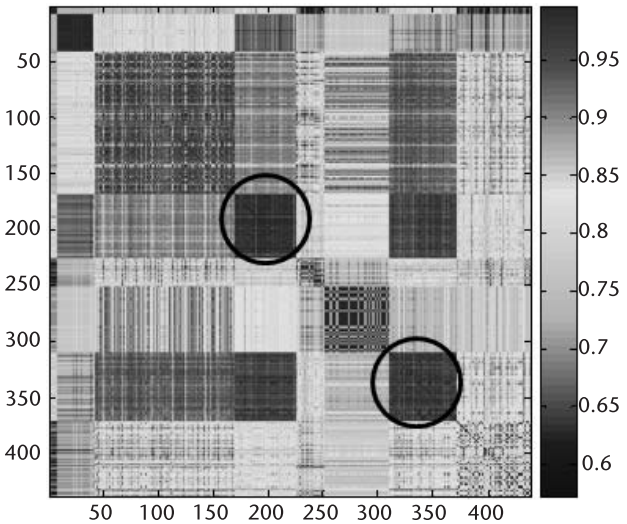


Figure 6.7 The pathway similarity matrix between the 438 *A. thaliana* genes. The matrix has been rearranged, using the clusters obtained by applying fuzzy *C*-means on the initial similarity matrix.

Table 6.3 The KEGG IDs for Three Candidate Pathways for Each of the 8 Clusters Found in the *A. Thaliana* Pilot Microarray Dataset

Cluster	*Size*	*Path 1 ID*	*Path 2 ID*	*Path 3 ID*
1	7	190	193	195
2	34	4130	53	3022
3	127	4710	230	280
4	56	940	600	903
5	25	3030	3022	3010
6	59	10	100	130
7	61	632	600	4130
8	69	280	290	770

gene memberships. We describe an evolutionary approach for pathway estimation, based on an ontological fuzzy rule system, in Section 6.4.2.

6.4.2 Mapping Genes to Pathways Using an OFRS in an Evolutionary Framework

6.4.2.1 The OFRS Format

The OFRS used in this application [8] has the following format:

$$\text{rule } i\text{: IF } gene\_group \text{ is } G_i \text{ THEN } pathway \text{ is } P_i \quad i \in [1, m]$$

where G_i is a set of genes represented by GO terms, and P_i is the KEGG pathway known to be associated with the given group. For example, the rule for the cell apoptosis pathway is

$$\begin{aligned} &\textbf{IF } gene\_group = \{BCL2, APAF1, CASP9, \cdots, FASL, FAS\} \\ &\textbf{THEN } pathway = \text{APOPTOSIS} \end{aligned} \tag{6.14}$$

More specifically, the above rule is used by replacing each gene by its GO representation, that is

IF *gene_goup*={("mitochondrial outer membrane," "antiapoptosis," "regulation of cellular pH,"...)$_{(BCL2)}$,("regulation of apoptosis," "nucleotide binding,"...)$_{(APAF1)}$, ("enzyme activator activity," "apoptotic program,"...)$_{(CASP9)}$...} ...**THEN** *pathway*=APOPTOSIS.

Although not done here, the above rule could be expanded conjunctively, using knowledge such as gene-gene interaction:

IF *gene_group* is G_{i1} AND *group_interaction* is G_{i2} **THEN** *pathway* is P_i,

the intuition being that the interaction between genes from the same pathway is higher than between those from different pathways.

The activation of rule i for a group of input (query) genes $\{q_j\}_{j=1,N}$ is computed as

$$a_i\left(\{q_j\}, rule_i\right) = \frac{\sum_{j=1}^{N} \max_{k=1,n} \left\{s\left(q_j, G_{ik}\right)\right\}}{N} \tag{6.15}$$

where $s(q_j, G_{ik})$ is the similarity between the jth input gene and the kth gene from pathway i, calculated using (6.2). When all the genes $\{q_j\}$ are explicitly mentioned in a given pathway, the rule activation given by (6.15) is 1. If a gene is not explicitly mentioned, however, the rule activation will reflect the degree of functional similarity between the unknown input gene and the genes from the given pathway. An activation lower than 1 also might be obtained if the GO and KEGG annotations for a given gene are different.

The input genes $\{q_j\}$ may belong to more than one pathway. Consequently, we need to address three problems: (1) find the number of pathways represented in the input group; (2) identify those pathways; and (3) assign the genes to the identified pathways. We can address all the above problems using an evolutionary C-means strategy. Assume that the genes $\{q_i\}$ belong to C pathways, which implies that the genes are split in C subgroups (clusters). The evolutionary C-means objective function J is designed to maximize the average membership of the C gene subgroups in pathways:

$$J(U) = \frac{\sum_{i=1}^{C} a_i\left(\{q_{n_i}\}\right)}{C} \tag{6.16}$$

where $\{n_i\}$ is a partition of N and a_i is the maximum activation of the ith gene subgroup, $\{q_{n_i}\}$, found across all the pathways. The pathways for which the maximum is obtained are selected as candidates for the output set. It is obvious that J is 1 when all C subgroups match perfectly in some pathways.

We can divide the input genes $\{q_j\}_{j=1,N}$ into two groups: one group $\{q_j\}^{\text{found}}$ that have the name represented in KEGG and another group $\{q_j\}^{\text{not-found}}$ that are not found in KEGG. In our experience, depending on the organism, the found gene category contains anywhere from 30% to 60% of the total number of input genes, N.

We can use two approaches to pathways mapping. The first approach consists of using only the input genes found in KEGG, $\{q_j\}^{\text{found}}$, to find the related pathways. We call this approach *crisp*, since it does not use the fuzzy gene matching based on the GO similarity. In the second approach, we map the input genes using their GO representation. As we mentioned earlier, this might require the use of automatic annotation software for genes without GO term association. We call this approach *fuzzy*, since the gene matching is based on their GO similarity.

6.4.2.2 Results

We used the July 2006 version of the KEGG pathway database for *H. sapiens* as our fuzzy rule database (same as in Section 6.4.1.3). Out of the 181 human pathways found in KEGG, we selected $m = 147$ of them that have at least 1 gene that is annotated with GO terms. The test dataset consisted of 10 sets of 15 genes each, randomly selected from KEGG pathways (3 pathways for each set) that have more than 20 genes.

In the first experiment, we compared the behavior of three gene-mapping methods, the crisp, the fuzzy, and the clustering-based method presented in Section 6.4.1, for a variable number of missing genes (not-found in KEGG) per pathway. The experiment consisted of randomly choosing a number of $p \in [0,5]$ genes from each gene subgroup (of the 3 existent in each of the 10 gene sets) and removing them from the KEGG pathways. This procedure artificially created not-found genes (with name not matched in KEGG) in the input set. For each gene set (of the total of 10), we ran the evolutionary C-means clustering algorithm with the objective function given by (6.16), a population of 50, a mutation rate of 0.2, and a number of 100 iterations. The pathway detection rate was calculated using (6.11). The results (the average detection rate for the 10 gene sets) are summarized in Table 6.4.

From Table 6.4, we see that when all genes (in number of 5, last column) in a subgroup are not-found, the crisp approach, obviously, cannot retrieve any pathway. However, if only 2 genes are known in each subgroup, the crisp method identifies the correct pathways in 90% of the cases. In addition, we see that the crisp approach is more stable in finding the right pathways than the fuzzy approach if at least 2 genes per pathway can be (crisply) mapped to KEGG.

At the same time, the fuzzy approach is able to identify, on average, 30% of the pathways, even in the case in which no gene from the input set could be mapped to KEGG (all were not-found). Obviously, the two methods based on ontological similarity are the only ones that can be used in such cases. It is also clear from Table 6.4 that our clustering approach presented in Section 6.4.1 cannot be used for mapping when many input genes are found in KEGG. Since the clustering-based method performs best when no information is available (last column in Table 6.4), however, it might be suitable for assigning genes to pathways in new genomes.

In conclusion, the crisp approach better identifies the pathways, while the fuzzy approach better maps the genes in the right pathway. This observation leads to a combined approach to gene mapping, in which the first step consists in using the crisp approach to map the found genes, followed by a second step in which the

Table 6.4 Average Detection Rate for Three Gene-Mapping Methods on the *H. Sapiens* Pathway Test Set at Different Numbers of Missing Genes per Pathway

Missing Genes/ Pathway	0	1	2	3	4	5
Crisp Detection Rate	0.93	0.93	0.9	0.9	0.6	0
Fuzzy Detection Rate	0.93	0.83	0.72	0.5	0.37	0.3
Clustering (Section 6.4.1) Detection Rate	0.5	0.5	0.57	0.57	0.6	0.63

remaining (not-found) genes are mapped, employing the fuzzy approach. The resulting OFRS-based algorithm for gene-to-pathway mapping is given in Algorithm 6.1 below.

Algorithm 6.1

Summary of the OFRS gene-to-pathway mapping algorithm.

Input:

-$\{q_i\}_{i=1,N}$
-KEGG rule base for the desired organism

Output:

-number C of active pathways
-the KEGG name of C active pathways
-a partition $\{q_i\}_{i\in n_c}, \underset{c=1,C}{\cup} n_c = N$ of the input genes.

Algorithm:

- find the genes represented in KEGG, $\{q_i\}^{\text{found}}$
-C=1
- WHILE $J < 1$ (6.17)
-C=C+1;
-map the $\{q_i\}^{\text{found}}$ genes to KEGG using the evolutionary C-means
-(ECM) procedure described in Section 6.4.2.1. For ECM we used:
-population size = 50;
-10 mutations/iteration;
-100 iterations;
- choose the partition $\{n_c\}$ with best J
-END WHILE
-map the rest of the genes $\{q_i\}^{\text{not-found}}$ one-by-one to the C pathways found above, using the ontological OFRS (6.15) and (6.16) based on their GO annotations (if not annotated, use an automatic annotation system [15–17] to do so.

We note that the evolutionary C-means procedure for mapping known genes to pathways, described in Section 6.4.2.1 might seem trivial at first glance. In fact, one could just enumerate the pathways in which the known input genes are found. Since a gene may appear in multiple pathways, however, for even a moderate number of input genes (e.g., 50) we would probably end up with all the pathways from KEGG as output. The key point of the evolutionary procedure is that it returns the minimal pathway set. Example 6.5 tries to clarify the problem.

Example 6.5 Consider the following input gene set: $\{q_i\}_{i=1,4}$={A, B, C, D}. The question is to find the minimal pathway set for the input genes, given the following pathway database:

IF {A, B} THEN P_1
IF {C, D} THEN P_2
IF {A, F} THEN P_3

IF {B, G} THEN P_4

The simplistic result for the mapping is $\{P_1, P_2, P_3, P_4\}$, since the four genes appear in all the pathway rules. However, the minimal set found by the evolutionary algorithm is $\{P_1, P_2\}$, which results in a $J = 1$ (6.17), C=2 and a partition {{A,B}, {C,D}}. Once $J = 1$ is attained, the search stops in order to prevent the apparition of the less desirable solution obtained for C=3: {{A},{B},{C,D}} that has a J = (0.5+0.5+1)/3=0.67.

6.5 Conclusion

In this chapter, we presented a fuzzy rule system that has memberships computed using ontological similarity, called the ontological fuzzy rule system (OFRS). The rules contained in the OFRS are obtained either from a curated database or by association rule mining. We presented an application of the OFRS for mapping genes to pathways. We investigated two approaches to the gene-mapping problem: one based on a disjunctive OFRS and clustering and the other based on a conjunctive OFRS and an evolutionary C-means procedure. The first approach is faster and may be suitable for finding pathways in new genomes in which little curated gene annotation is available. The second approach may work best for mapping new genes in well-annotated genomes.

Acknowledgments

We would like to thank Trupti Joshi and Erik Taylor for their technical assistance.

References

[1] Andreasen, T., H. Bulskov, R. Knappe, "On Ontology-Based Querying," H. Stuckenschmidt (ed.), *18 th Int. Joint Conf. on Artificial Intelligence*, Acapulco, Mexico, August 9, 2003, pp. 53–59.

[2] Baader, F, et al., *The Description Logic Handbook*, Cambridge, U.K.: Cambridge University Press, 2003, p. 578.

[3] Bison, G., "Learning in FOL with a Similarity Measure," *AAAI Symposium*, San Jose, California, July 12–16, 1992, pp. 82–97.

[4] Buchanan, B. G., and E. H. Shortliffe (eds.), *Rule-Based Expert Systems*, Menlo Park, CA: Addison-Wesley Publising Co., 1984.

[5] Doniger, S.W., et al., "MAPPFinder: Using Gene Ontology and GenMAPP to Create a Global Gene-Expression Profile from Microarray Data," *Genome Biology*, Vol. 4, No. 1, 2003, p. R7.

[6] Klir, G. J., and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Upper Saddle River, NJ: Prentice Hall PTR, 1995, p. 574.

[7] Khan, S., et al., "GoFigure: Automated Gene Ontology Annotation," *Bioinformatics*, Vol. 19, No. 18, 2003, pp. 2484–2485.

[8] Popescu M., and D. Xu, "Mapping Genes to Pathways Using Ontological Fuzzy Rule Systems," *IEEE Int. Fuzzy Systems Conf.*, London, U.K., July 23–26, 2007.

[9] Popescu M., D. Xu, and E. Taylor, "GoFuzzKegg: Mapping Genes to KEGG Pathways Using an Ontological Fuzzy Rule System," *Proc. on IEEE Symp. on Comp. Int. in Bioinf. and Comp. Biol.*, Honolulu, Hawaii, April 1–5, 2007, pp. 298–303.

[10] Bezdek, J. C., *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York: Plenum, 1981, p. 272.

[11] Hathaway, R. J., Bezdek, J. C., "NERF c-Means: Non-Euclidean Relational Fuzzy Clustering," *Pattern Recognition*, Vol. 27, No. 3, 1994, pp. 429–437.

[12] Popescu, M., et al., "A New Cluster Validity Measure for Bioinformatics Relational Datasets," *World Congress on Computational Intelligence, WCCI2008*, Hong Kong, June 1–6, 2008, pp. 726–731.

[13] Perez, A.J., G. Thode, and O. Trelles, "AnaGram: Protein Function Assignment," *Bioinformatics*, Vol. 20, No.2, 2004.

[14] Prlic, A., et al., "WILMA-Automated Annotation of Protein Sequences," *Bioinformatics*, Vol. 20, No. 1, 2004.

[15] Reich, M., et al., "GenePattern 2.0," *Nat. Genet.,* Vol. 38, No. 5, 2006, pp. 500–501.

[16] Tomfohr, J., J. Lu, and T. B. Kepler, "Pathway Level Analysis of Gene Expression Using Singular Value Decomposition," *BMC Bioinformatics*, Vol. 6, No. 225, 2005.

[17] Van Eck, N. J., et al., "Visualizing the Computational Intelligence Field," *IEEE Comp. Intell. Mag.*, Vol. 1, No. 4, 2006, pp. 6–10.

[18] Xu, D., et al., *Applications of Fuzzy Logic in Bioinformatics*, London: Imperial College Press, 2008, p. 225.

[19] Yager, R., "On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decision Making," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 18, No. 1, 1988, pp. 183–190.

[20] Zadeh, L., "From Computing with Numbers to Computing with Words," *Applied Math and Computer Science*, Vol. 12, No. 3, 2002, pp. 307–324.

CHAPTER 7

Extracting Biological Knowledge by Association Rule Mining

F. Javier Lopez, Armando Blanco, Fernando Garcia, and Carlos Cano

The availability of the complete genome from diverse species and the advent of high-throughput genomic technologies have generated a great amount of structural and functional genomic information, boosting bioinformatics research to develop computational techniques that help to analyze such a huge amount of data [1]. In this context, association rules have emerged as a powerful tool to analyze biological data, due to their ability to manage large datasets, their capacity to treat heterogeneous information, and the intuitive interpretation of the results obtained with this technique. Thus, association rules have been widely used in bioinformatics, their applications spanning from pure data-mining approaches to signaling-pathways inference, protein-protein interaction prediction, or regulatory modules discovery [2, 3, 4, 5].

The search for a formal methodology to organize, present, and computationally manage biological data has recently given ontologies a major role in bioinformatics. One of the most popular bio-ontologies is the Gene Ontology (GO) [6]. The Gene Ontology Project aims to satisfy the need for consistent descriptions of gene products in different databases. It has become the de facto standard that provides a structured, controlled vocabulary for describing the roles of genes and gene products in many organisms (Chapter 1).

This chapter presents methodologies for association analysis based on association rule mining and discusses several applications in bioinformatics, mainly focused on GO and microarray analysis. The chapter is organized as follows: Section 7.1 overviews the main aspects of association rules and fuzzy association rules. Section 7.2 describes some applications of association rules involving the Gene Ontology, and Section 7.3 presents a set of applications for microarray analysis.

7.1 Association Rule Mining and Fuzzy Association Rule Mining Overview

In 1993, Agrawal proposed an algorithm for extracting association rules from large databases [7]. The initial application of association analysis techniques was the

study of the hidden relations in *market-basket databases*. Typically, these databases contain information about the products bought by the customers in each purchase. Thus, a market-basket database consists of a set of transactions, each of them containing the items acquired in that transaction (Table 7.1). Hence, the main objective when an association analysis is carried out over this kind of databases is to obtain relations of the form

$$\{Milk\} \rightarrow \{Butter\}$$

This is basically an *association rule*, and represents the expression: *Those who buy milk also buy butter.*

This type of information may be of great interest for a supermarket administrator, since, for example, sales might be increased by placing certain products together.

Association rules have also been successfully applied in many other different fields, including Web mining, advertising, bioinformatics, and so on. Moreover, since they were first proposed in 1993, they have become one of the main techniques for knowledge discovery in databases (KDD).

7.1.1 Association Rules: Formal Definition

Let $I = \{x_1, x_2, \ldots, x_n\}$ be a set of attribute-value pairs or *items*. Let D be a transactional database, in which each *transaction* is a set of items $T \subseteq I$. An *association rule* is an expression of the form $X \rightarrow Y$, where X and Y are sets of items (or *itemsets*) so that $X \cap Y = \emptyset$. The itemset X is called the *antecedent* of the rule, while Y is called the *consequent*. An association rule like this indicates that if X occurs, then Y is likely to occur. The probability that Y occurs, given that X has occurred is called the *confidence* of the rule. The probability that both X and Y will occur is called the *support* of the rule. Thus, classical association rule mining algorithms aim to extract association rules with support and confidence greater than some user-specified threshold.

A transaction T is said to *support* an itemset $X \subseteq I$, if $X \subseteq T$, or, in other words, T contains all the items in X. Thus, the *support* of an itemset X is the percentage of transactions in the database that supports X, or, in other words, the probability of finding the itemset X in the database. Therefore, the support of a rule $X \rightarrow Y$ can be calculated as

$$Supp(X \rightarrow Y) / Supp(X \cup Y)$$

Table 7.1 Example of a Market-Basket Database

Transaction	*Items*
1	{Bread, Milk, Butter}
2	{Beer, Eggs, Milk, Butter, Fruit}
3	{Milk, Butter}
…	…

and the confidence of a rule $X \rightarrow Y$ can be defined as

$$Conf\left(X \rightarrow Y\right) = Supp\left(X \rightarrow Y\right) / Supp\left(X\right)$$

Finally, an itemset X is said to be *frequent* if its support is greater than some user-specified threshold.

For example, consider the information in Table 7.2 which contains some structural data for a set of yeast genes. This table can be easily seen as a transactional database, in which each row represents a transaction, and the attributes in each column form the items of the transaction. (Table 7.3).

Consider now the itemset Z:

Table 7.2 An Example of a Data Table

Gene	*Gene Length*	*Intergenic Length*	*Gene Orientation*
YAL002W	LARGE	LARGE	TANDEM
YAL003W	SHORT	LARGE	DIVERGENT
YAL008W	SHORT	SHORT	TANDEM
YAL009W	MEDIUM	SHORT	DIVERGENT
YAL010C	MEDIUM	SHORT	DIVERGENT
YAL011W	MEDIUM	MEDIUM	TANDEM
YAL012W	MEDIUM	MEDIUM	TANDEM
YAL013W	MEDIUM	MEDIUM	DIVERGENT
YAL015C	MEDIUM	MEDIUM	TANDEM
YAL017W	LARGE	LARGE	DIVERGENT
YAL018C	MEDIUM	LARGE	DIVERGENT
YAL019W	LARGE	SHORT	TANDEM
YAL021C	LARGE	MEDIUM	TANDEM

Table 7.3 Table 7.2 Transformed into a Transactional Data Table

Transaction	*Items*
YAL002W	{(Gene length = LARGE), (Intergenic length = LARGE), (Gene orientation = TANDEM)}
YAL003W	{(Gene length = SHORT), (Intergenic length = LARGE), (Gene orientation = DIVERGENT)}
YAL008W	{(Gene length = SHORT), (Intergenic length = SHORT), (Gene orientation = TANDEM)}
…	…

$$Z = \begin{Bmatrix} (Gene\ length = MEDIUM), (Intergenic\ length = MEDIUM), \\ (Gene\ orientation = TANDEM) \end{Bmatrix}$$

This itemset is supported by transactions (genes) *YAL011W, YAL012W, and YAL015C,* and there are 13 transactions in total, therefore *Supp(Z)=3/13=0.231*. Consider now the association rule:

$$R = \{(Gene\ length = MEDIUM), (Intergenic\ length = MEDIUM)\} \rightarrow \{(Gene\ orientation = TANDEM)\}$$

The support of *R* is

$$Supp(R) = Supp(Z) = 0.231$$

and the confidence of *R* can be calculated as

$$Conf(R) = Supp(R) \Big/ Supp\left(\begin{Bmatrix} (Gene\ length = MEDIUM), \\ (Intergenic\ length = MEDIUM) \end{Bmatrix} \right) = (3/13)/(4/13) = 0.75$$

The main drawback of association rule mining techniques is that the number of generated rules is often large, many of them providing redundant or nonrelevant information. The support/confidence framework has been proven to be insufficient to deal with this problem. Therefore, additional strategies and interestingness measures have been proposed to enhance the interpretability of the resultant rule set. However, pattern *interestingness* is often confused with pattern *accuracy*. The majority of the literature focuses on maximizing the accuracy of the discovered patterns, ignoring other important quality criteria. In fact, the correlation between accuracy and interestingness is not so clear. For example, the statement "men do not give birth" is highly accurate, but not interesting at all [8]. Hence, there is not a widespread agreement on a formal definition for the interestingness of a rule. Some authors have even defined the interestingness of a pattern as a compendium of concepts, such as *conciseness*, *coverage*, *reliability*, *peculiarity*, *diversity*, *novelty*, *surprisingness*, *utility,* and *actionability* [9]. Thus, many rule interestingness measures and rule reduction strategies have been proposed (for a review, see [9, 10]).

Summarizing, the association rule mining process is generally divided into two steps:

1. Finding the set of frequent itemsets. The majority of association mining research effort has been focused on this step, since it is the most computationally expensive phase.
2. Deriving association rules with confidence greater than a user-specified threshold from the frequent itemsets.

7.1.2 Association Rule Mining Algorithms

A great number of algorithms have been proposed for association rule mining. So far, there is no published implementation that outperforms every other implementation on every database with every support threshold [11]. Classical association rule mining algorithms can be divided into two major categories, which correspond to two main strategies for finding *valid* (i.e., frequent) itemsets: *candidate generation* and *pattern growth* algorithms. The majority of the classical algorithms are of candidate-generation type [7, 12, 13]. This type of algorithms generates sets of *candidate* itemsets that are then validated following the imposed constraints (e.g., support ≥ min support threshold). Furthermore, the generation of candidate itemsets is based on previously identified valid itemsets. The main algorithms of this type are the well-known Apriori and Eclat [7, 12].

Unlike candidate-generation algorithms, pattern-growth methods avoid candidate generation by constructing complex data structures that concisely store the relevant information in the dataset. So long as the data structure fits in memory, no more dataset accesses are necessary, once it has been populated. A number of algorithms of this type have been proposed, the most popular one being the Frequent-Pattern Growth (FP-Growth) algorithm [14, 15].

Further divisions within both classes (candidate generation and pattern growth) are based on the strategy to traverse the search space (deep-first or breadth-first), and on the different data structures they use (hash-trees, enumeration-set trees, prefix trees, FP-trees, H-struct, and so on).

In addition, another type of algorithm, derived from the fundamental ones, has been proposed. The aim of these methods is to generate a condensed rule set from which all the rules can be derived, thus optimizing the valid itemset search procedure. Moreover, the obtained rule set is smaller than the complete rule set, facilitating, in this way, the interpretation of the rules [16–19].

Moreover, it is worth mentioning the effort carried out to develop methodologies to extract information from transactions where a taxonomy is defined on the items [20–22]. Figure 7.1 shows an example of a taxonomy. This type of methodology may be of special interest in this book, since an ontology can be viewed as a

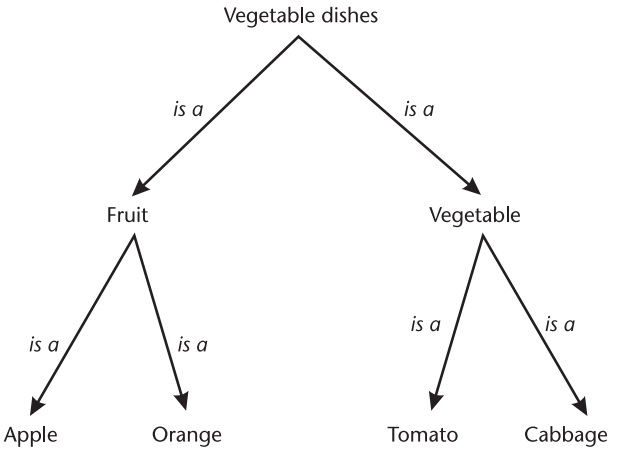


Figure 7.1 An example of a taxonomy.

taxonomy, as it represents a set of concepts hierarchically organized according to their specificity.

In summary, a great number of algorithms have been proposed for association rule mining, the main ones being Apriori, Eclat, and FP-growth. In general, many implementations are based on Apriori and FP-growth. Apriori is the most popular, and it is usually the one chosen when using association rule mining in any application. Section 7.1.3 describes in detail the basic Apriori algorithm. Many improvements of this algorithm have been proposed, the most efficient ones being those described in [23, 24]. A more comprehensive listing and description of association rule mining algorithms can be found in [10, 25, 26].

7.1.3 Apriori Algorithm

As it was previously stated, Apriori divides the association rule extraction process into two well-differentiated steps: (1) Finding the sets of frequent itemsets; and (2) deriving the association rules from the frequent itemsets. Let us start by describing the first step. Apriori takes advantage of the *antimonotone* property of the support measure to reduce the search space.

- Every subset of items of a frequent itemset is also frequent. That is, given a frequent itemset X, for all $Y \subseteq X$, Y is also a frequent itemset.
- Due to the previous property, if an itemset X is not frequent, any other itemset containing X cannot be frequent. That is, given a nonfrequent itemset X, for all $Z \supseteq X$, Z is not frequent.

The basic idea is to generate the set of k *itemsets* (itemsets of k elements), by combining the frequent (k–1) itemsets. The first step is to identify frequent items by carrying out a first scan of the complete data table to count the number of times that each item appears in it. Nonfrequent items are discarded, obtaining a set of frequent items (F_1). The process continues as follows: those pairs of frequent (k–1) itemsets sharing their first k–2 items, but with a different k–1 item (i.e., the last item in the itemset), are combined. The result of the combination is a k itemset in which the k–2 first items correspond to the k–2 first items of the combined itemsets, and items k–1 and k are the last items of the combined itemsets (i.e., the k–1 items of the combined itemsets). A schematic representation of the procedure is shown in Figure 7.2.

Note that to successfully carry out this procedure, it is necessary to have previously set an order relation between the items. Thus, in the example of Figure 7.2, for every itemset containing *ItemA* and *ItemB*, *ItemA* will always appear before *ItemB* when listing the items in the itemset. Similarly, for every itemset containing *ItemB* and *ItemC*, *ItemB* will always appear before *ItemC* when listing the items in the itemset, and so on. The order relation between the items is essential to ensure that every potential frequent itemset is considered.

Once the set of *candidate* k itemsets are obtained (*candidate* since they are not necessarily frequent), another scan of the data table is needed to calculate their sup-

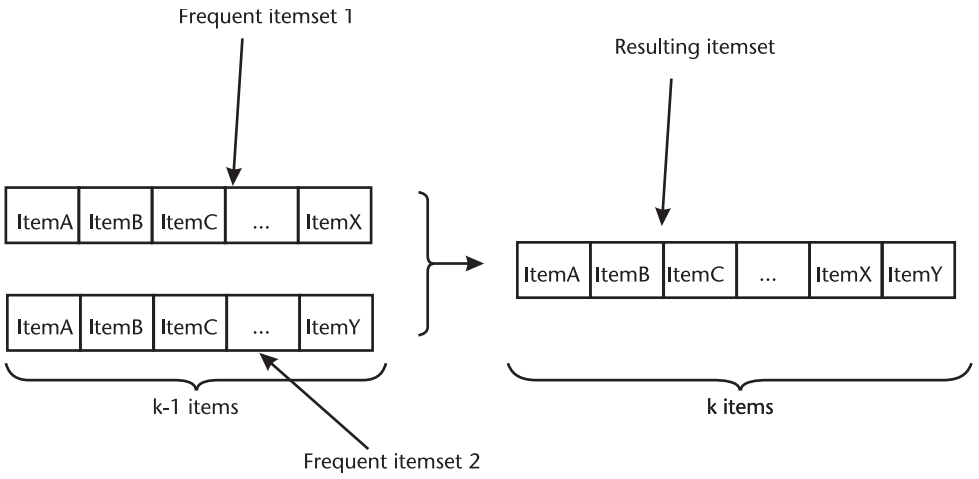


Figure 7.2 An example of the combination of two (*k*-1) itemsets to obtain a *k* itemset.

port. Then, the nonfrequent *k* itemsets are discarded, obtaining the set F_k. Figure 7.3 shows the pseudocode of the procedure described in this section.

Association rules are derived from each itemset, once the complete set of frequent itemsets is obtained (second phase of the algorithm). This phase is common to every association rule mining algorithm, since the variations from one procedure to another reside mostly in the way they get the list of frequent itemsets. Given an itemset, the idea is to create a consequent for each possible subset of items in the itemset. The rest of the items in the itemset that are not in the consequent constitute the antecedent of the rule, and thus, in principle, a rule is generated from each subset. Furthermore, the efficiency of the process can be improved, since it is not necessary to consider every possible subset of items. Figure 7.4 shows the procedure to efficiently derive rules from a given itemset.

```
1. k = 1                                          //Find every frequen 1-itemset, i.e., the set
2. F_k = {i| i is an item and support(i) ≥ minSupp}   // of frequent items

3. Repeat
4. k = k + 1                                      //The set of candidate k-items is generated
5. C  = candidateGeneration(F_{k-1})              //by combining the frequent (k-1)-itemsets

6. For each transaction t
7.   C_t = set of k-items in C_k contained in t
8.   For each candidate c in C_t                  //The count of itemset c is increased
9.        count(c) = count(c) +1
10. For each candidate c in C_k
11.       supp(c) = count(c)/total_number_of_transactions   //Only the frequent itemsets are selected
12. F_k ={d|d ∈ C_k and supp(d) ≥ minSupp}
13. Until F_k = ∅
14. The result is obtained as the union of the F_k sets.
```

Figure 7.3 Procedure for frequent itemset generation. Note that function candidateGeneration(F_{K-1}) combines every frequent itemset in F_{k-1} to obtain the *candidate k*- temsets, as described in Section 7.1.3.

```
Function generateRules(itemset)

2.- List_of_consequents_1 = set of 1-itemsets, each of them formed by one item in "itemset"
3.- m = 1                    // size of the consequents, initially only 1 attribute
4.- ruleList = []

5.- While List_of_consequents_m ≠ ∅ and the size of the itemset > m:
6.-    List_of_consequents_m+1 = generateConsequents(List_of_consequents_m)
7.-    For each consequent cq in List_of_consequents_m+1:
8.-       confidence = support(itemset) / support(itemset - cq)
9.-       If confidence ≥ confidence_threshold:
10.-         ruleList = ruleList + [(itemset – cq) → cq]
11.-      Else
12.-         remove cq from List_of_consequents_m+1
13.-   List_of_consequents_m = List_of_consequents_m+1
14.- Return ruleList
```

Figure 7.4 Procedure for deriving rules from a given itemset. Note that the function consequentsGeneration(H_m) combines every consequent of size m to obtain the consequents of size $m + 1$. It does exactly the same as the function candidateGeneration() in Figure 7.2.

7.1.4 Fuzzy Association Rules

Classical crisp association rule mining algorithms partition continuous domains to deal with continuous attributes. For example, consider the data in Table 7.4. Attributes *gene length* and *intergenic length* are continuous; therefore, it is infeasible to look directly for frequent itemsets that involve these two attributes. A preprocessing step is needed to discretize both domains, or, in other words, to partition the continuous domains in intervals. After the partition is carried out, each continuous value is replaced by the interval to which it belongs. Several strategies have been proposed for discretizing the continuous domains [27, 28].

Nevertheless, when dividing an attribute into intervals covering certain ranges of values, *the sharp boundary problem* arises. Elements near the boundaries of a crisp set (interval) will be either ignored or overemphasized. For example, rules like "If the gene length is in the interval [1541, 14733], then the G+C content tends to be in the interval [0.26, 0.38]," and "Large genes tend to have low G+C content" may all be meaningful depending on different situations. While the former is more specific and the latter is more general in semantic expressions, however, the former presents the previously called *sharp boundary problem*, or, in other words, genes of 1540 bp and with 0.25 of G+C content may not be considered. In contrast, the latter is more flexible and can reflect these boundary cases [29]. Moreover, fuzzy set theory has been proven to be a superior technology to enhance the interpretability of these intervals [30]. Hence, in the fuzzy case, continuous domains are fuzzified by partitioning them into fuzzy sets (Figure 7.5). Therefore, *fuzzy* association rules

Table 7.4 An Example of a Data Table

Gene	*Gene Length*	*Intergenic Length*	*Gene Orientation*
YAL002W	2217	546	TANDEM
YAL003W	1290	742	DIVERGENT
YAL008W	492	280	TANDEM
YAL009W	4299	188	DIVERGENT
YAL010C	1965	188	DIVERGENT
YAL011W	1107	215	TANDEM
YAL012W	918	268	TANDEM
YAL013W	471	250	DIVERGENT
YAL015C	2634	250	TANDEM
YAL017W	330	149	DIVERGENT
YAL018C	885	683	DIVERGENT
YAL019W	393	683	TANDEM
YAL021C	1215	99	TANDEM

are also expressions of the form: $X \rightarrow Y$, but in this case, X and Y are sets of fuzzy attribute-value pairs.

The traditional way to determine fuzzy sets is to consult a domain expert who defines the membership functions. However, this requires access to domain knowledge, which can be difficult or even impossible to acquire. Clustering, genetic algorithms, and so fourth, in this case are used for the definition of the fuzzy sets. Therefore, it is the methodology for defining the fuzzy sets that is based on clustering,

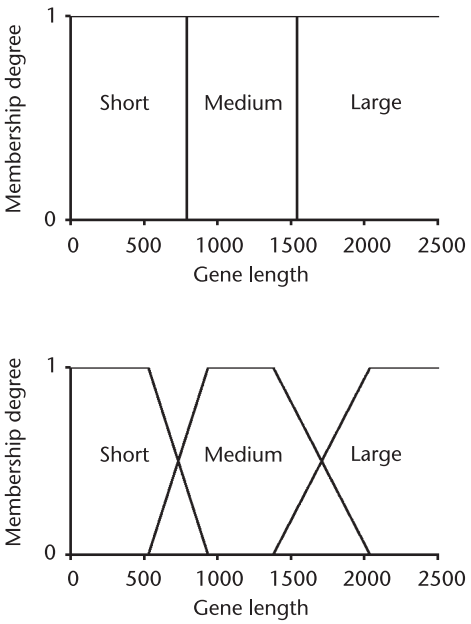


Figure 7.5 An example of crisp and fuzzy partitions.

genetic algorithms, and so fourth, and not the rules. Thus, several approaches have been proposed for automatically defining fuzzy sets, for example, approaches based on clustering [31–33], on genetic algorithms [34], and so on. Although these strategies could be useful in certain cases, however, they should be used carefully, since the obtained fuzzy sets could be hard to fit to meaningful labels.

When assessing a fuzzy association rule, the usual approach consists of using the fuzzy counterparts of the support and confidence measures. Several generalizations of these two measures have been proposed [35]. The standard approach is to replace the set-theoretic operations by their corresponding fuzzy set-theoretic operations. Thus, given a transactional database D, the membership degree of a transaction $t \in D$ to a fuzzy itemset X is calculated as $X(t) = \otimes_{Xi \in X} X_i(t)$, where $\otimes$ represents a *t-norm* [36]. A so-called *t-norm* $\otimes$ is a generalized logical *conjunction*, or, in other words, a function $[0,1] \times [0,1] \rightarrow [0,1]$ which is associative, commutative, and monotone increasing, and which satisfies

$$a \otimes 0 = 0,$$
$$a \otimes 1 = a, \text{ for all } 0 \leq a \leq 1$$

Common examples of *t*-norms are

$$\text{minimum}(a,b) = \min(a,b),$$
$$\text{product}(a,b) = a \cdot b,$$
$$\text{Lukasiewicz t-norm}(a,b) = \max(a+b-1,0)$$

Thus, *t*-norms are used for defining the *intersection* of fuzzy sets. Given two fuzzy sets defined over a domain Z and their corresponding membership degree functions $A{:}Z \rightarrow [0,1]$, $B{:}Z \rightarrow [0,1]$, the intersection of the two fuzzy sets, $A \cap B$, is defined as follows:

$$(A \cap B)(z) = A(z) \otimes B(z), \text{ for all } z \in Z$$

For example, consider the fuzzy itemset:

$$X = \{(Gene\ length = LARGE)(Intergenic\ length = SHORT)\}$$

and the transaction:

$$YAL010C = \begin{Bmatrix} (Gene\ length = 1965), (Intergenic\ length = 188), \\ (Gene\ orientation = DIVERGENT) \end{Bmatrix}$$

Suppose that the membership degree of 1965 to the fuzzy item *Gene length = LARGE* is 0.6, and that the membership degree of 188 to the fuzzy item *Intergenic length = SHORT* is 1. Also consider that the chosen *t*-norm is the *minimum*. Then, the membership degree of transaction *YAL010C* to the fuzzy itemset X is given by

$$X(YAL010C) = \min(0.6, 1) = 0.6$$

Hence, considering all of this, the fuzzy support of an itemset X is usually defined as

$$Supp(X) = \sum_{t \in D} \left[\otimes_{Xi \in X} X_i(t) \right]$$

that is, the sum of the membership degrees of the transactions in the database to the itemset X. Finally, the fuzzy support and confidence of a fuzzy association rule $X \rightarrow Y$ is given by

$$Supp(X \rightarrow Y) = \sum_{t \in D} X(t) \otimes Y(t),$$
$$Conf(X \rightarrow Y) = \left[\sum_{t \in D} X(t) \otimes Y(t) \right] \Big/ \sum_{t \in D} X(t)$$

Even though the majority of fuzzy proposals are based on the fuzzy extensions described above, some alternative approaches have been reported [30, 37, 38].

The development of efficient algorithms for fuzzy association rule mining has been paid little attention. This might be explained by the fact that, in general, standard crisp algorithms can be adapted for extracting fuzzy association rules in a straightforward way [37, 39]. The first proposal for fuzzy association rule mining was reported in [40]. The authors presented a straightforward approach in which a membership threshold is fixed for transforming fuzzy transactions into crisp ones before running an ordinary association rule mining algorithm. After this, some other authors presented algorithms for fuzzy association rule mining such as F-APACS and FARM [41, 42], extensions of the Equi-depth (EDP) algorithm [43], and other Apriorilike methods [32, 44]. For a more extensive listing, please refer to [30]. Finally, the problem of mining association rules in fuzzy taxonomies has also been addressed in many papers [45, 46]. A fuzzy taxonomy is a hierarchically structured set of items that reflects partial belongings among items on different levels (Figure 7.6).

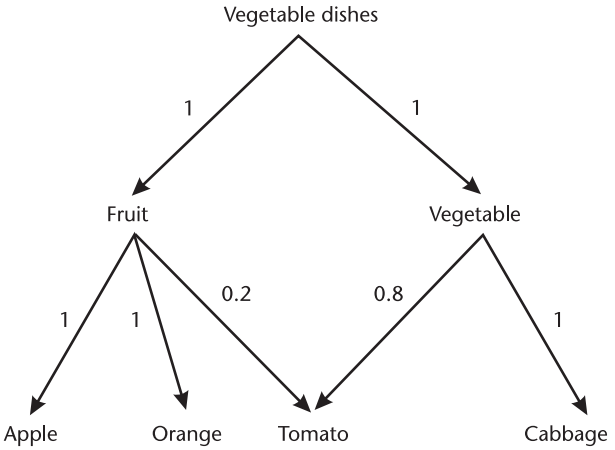


Figure 7.6 An example of fuzzy taxonomy.

7.2 Using GO in Association Rule Mining

As already mentioned in the introduction of this chapter, association rules have been widely applied in bioinformatics. In many of these applications, the Gene Ontology (GO) plays a major role. This section reviews some of the strategies proposed so far to combine GO and association rule mining, which could be categorized as follows:

1. Works which integrate GO structured information with other data sources in order to obtain rules relating GO terms and the rest of the variables;
2. Works which make use of GO annotations to biologically validate the obtained rule sets;
3. Other joint applications of association rules and GO, such as the development of classifiers, which aim to automatically annotate genes or gene products based on other features, and so on.

7.2.1 Unveiling Biological Associations by Extracting Rules Involving GO Terms

Some authors incorporate GO terms into their datasets to obtain associations that relate the terms with their studied variables. Rules involving GO terms are able to describe, in an intuitive and concise way, relations between biological concepts and the rest of the studied variables. This makes the integration of GO terms with other data sources an attractive approach, and thus, several authors have recently developed different proposals [2, 3, 47].

In these types of studies, the dataset typically consists of a data table in which rows represent genes and columns represent the set of variables of interest (e.g., microarrays, annotations from other databases, gene features, and so on). An example of this type is the work reported by Carmona-Saez et al. [3]. Thus, the naive approach to integrating the GO terms in the analysis consists of directly including the GO annotations of each gene in an additional column. For example, in Table 7.5, an additional column has been added containing the list of GO annotations for each gene. Each GO term constitutes an item of the form (*GO annotation* = *GO:xxxxxxx*). Thus, in running an association rule mining algorithm over this data table, associations between the GO terms and the rest of variables might be obtained. Since the number of terms is quite high (~6,800 terms related with the human genome), and in these types of studies, associations among GO terms are not usually of interest, the search space may be substantially pruned by avoiding generating itemsets containing more than one GO annotation.

Nevertheless, when using the terms in which the genes are directly annotated, some problems might arise:

1. Some of these terms may represent very specific concepts. This means that only few genes would be annotated in these terms, and thereby, these terms would not form frequent itemsets.
2. Suppose a set of genes annotated to a term T and a different set of genes annotated to a term T', where T' is an ancestor of T. When counting the

Table 7.5 An Example of a Data Table in Which GO Terms Have Been Included

Gene	*Gene Length*	*Intergenic Length*	*Gene Orientation*	*GO Annotations*
YAL002W	LARGE	LARGE	TANDEM	GO:0045324,GO:0033263, GO:0005624,GO:0003674
YAL003W	SHORT	LARGE	DIVERGENT	GO:0006414,GO:0005853, GO:0005840,GO:0003746
YAL008W	SHORT	SHORT	TANDEM	GO:0008150,GO:0005741, GO:0005739,GO:0003674
YAL009W	MEDIUM	SHORT	DIVERGENT	GO:0030437,GO:0007126, GO:0006997,GO:0016021, GO:0042175,GO:0004721
YAL010C	MEDIUM	SHORT	DIVERGENT	GO:0000002,GO:0000001, GO:0007005,GO:0006461, GO:0045040,GO:0000723, GO:0032865,GO:0005741, GO:0001401,GO:0003674
...	...	...	...	...

occurrences of the itemsets containing T' in the data table, those genes annotated to T would not be taken into account, since only the term T appears in their transactions. Since terms are considered to share the attributes of all the parent nodes, all the genes annotated to term T must also be taken into account when counting the frequency of term T', otherwise an important loss of information might occur.

Martinez et al. [47] avoided these problems by including in the data table not only the terms in which the genes are directly annotated, but also all of their ancestors. However, an important drawback arises when using this last strategy: if every ancestor is included in the analysis, very general terms (e.g., *molecular_function, biological_process, cellular_component,* and so on) may be considered. These terms are so general that do not provide any interesting information. Moreover, they slow down the mining process and disturb the interpretation of the final rule set, since they generate many trivial or uninteresting rules.

Hence, a possible approach consists of including only terms of a selected GO level. Those terms below the selected depth are mapped to the corresponding one in that level, and those above are discarded. Some applications (not necessarily association rule-based applications), such as FatiGO [48], adopted this methodology, and, in principle, it seems that GO level 3 represents a good compromise between information quality and the number of annotated genes [49]. Nevertheless, GO levels are not homogeneous, or, in other words, the terms representing general concepts and others that represent more specific concepts might be found in the same GO level [50]. Therefore, some information might be lost when using this strategy.

Lopez et al. [2] noticed the previous problems and proposed an alternative methodology: consider all the ancestors, calculate the information provided by each term, and remove those that are uninformative. By assuming that the more specific a term is, the more information it gives, the *information content* (IC) of a term T can be computed as $IC(T) = -log(P(T))/ -log(P(min))$, where $P(T)$ represents the probability of finding T or a child of T in the ontology. The denominator is used to normalize, or, $P(min) = 1/Total\_number\_of\_annotations$. Note that the deeper the GO term is in the ontology, the greater its IC. This is due to the ontological structure of GO. If the number of annotations decreases, the probability of the terms occurring also decreases, and therefore their IC tends to increase (Figure 7.7).

Additionally, if many rules involving GO terms are obtained, these authors propose to reduce the resultant rule set by merging subsets of rules containing GO terms that may provide similar information. This strategy takes advantage of the GO structure to filter the rule set. First of all, a scan of the rule set is carried out to look for groups of rules involving a GO term and sharing all their items except the GO node. For each group, if there is a GO term in it that is a common ancestor for the rest of the GO nodes in this rule set, only the rule involving the common ancestor is maintained, while the rest of rules in the group are discarded. This strategy relies on the idea that each Gene Ontology term shares the attributes of all its parent nodes. Since it is ensured that the terms included in the analysis are informative

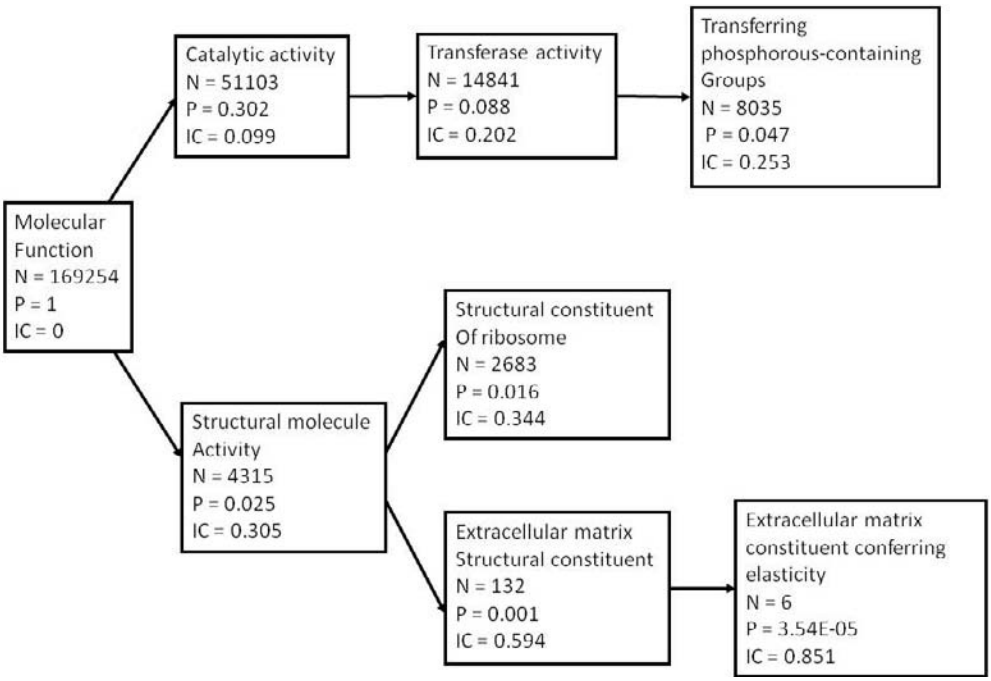


Figure 7.7 A fragment of the ontology *molecular function* in GO. Each node is labeled with its name, the number of annotations in it, and under it (*N*), the probability derived from the number of annotations (*P*) and its information content (IC). The *Total_number_of_annotations* used to calculate the probabilities corresponds to the number of annotations of the highest node in the ontology, or 169,524 in this case.

enough by setting an appropriate IC threshold, the common ancestor represents the most intuitive term. See Figure 7.8 for an example.

Regarding the application of fuzzy techniques, to the extent of our knowledge, so far only the work by Lopez et al. [2] makes use of fuzzy association rules in its study. In this case, the domains of the continuous variables are partitioned into three fuzzy sets that represent the linguistic labels *HIGH*, *MEDIUM*, and *LOW*. Fuzzy sets are defined by using the expert-guided percentiles p_{20}, p_{40}, p_{60}, p_{80}, as shown in Figure 7.9, and a fuzzy version of the Top Down FP-Growth algorithm [51] is used to mine the data table.

It is worth mentioning the absence of works that, trying to extract useful knowledge from the Gene Ontology by association rule mining, consider GO as a taxonomy. As previously stated, an ontology can be considered as a taxonomy, since it represents a set of concepts hierarchically organized, according to their specificity. Many works have proposed efficient algorithms for mining association rules from taxonomies, and their application in future works may provide higher quality rule sets. In addition, the use of algorithms able to mine fuzzy taxonomies could also be interesting. However, their application does not make sense as long as there is no fuzzy version of the Gene Ontology.

7.2.2 Giving Biological Significance to Rule Sets by Using GO

Association rule discovery is a *nonsupervised* data-mining technique. This means that, in principle, there is no a priori knowledge to compare it with the resultant rule set and quantify its goodness, since the objective is to unveil unknown patterns.

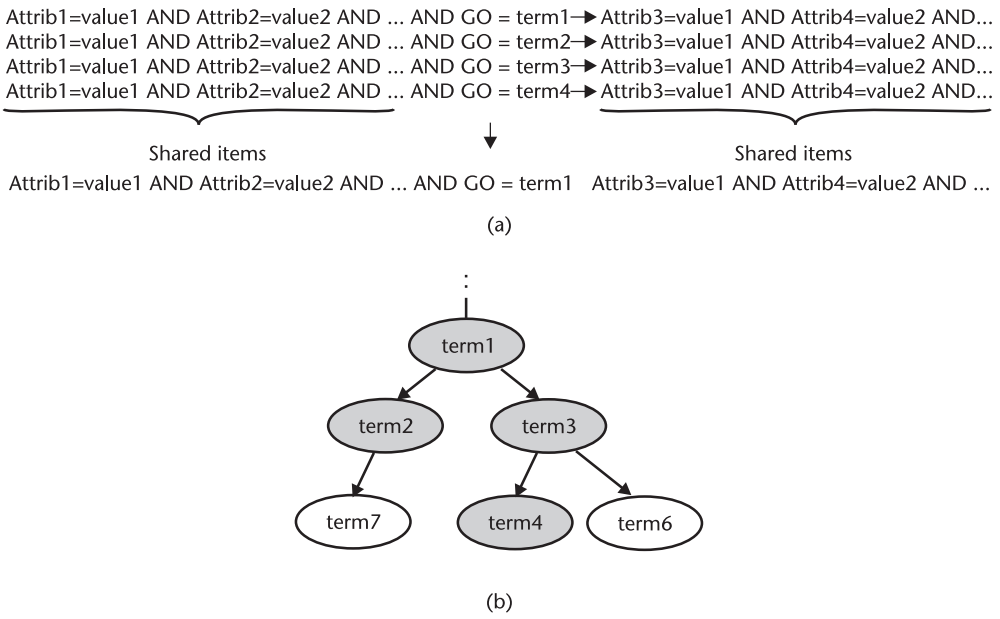


Figure 7.8 This figure shows an example in which four rules are merged into only one. (a) shows a group of 4 rules sharing all their items except the one involving the GO term. These 4 rules are merged into the more general one. (b) shows the distribution of the terms in the ontology.

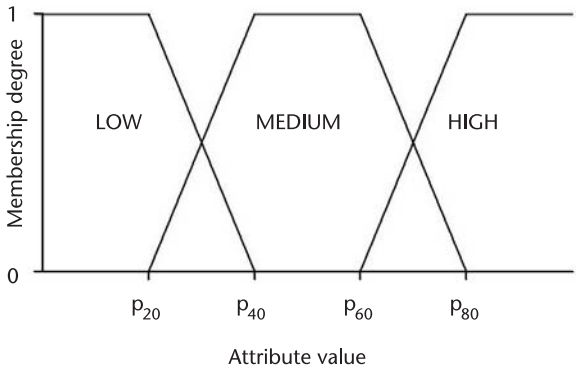


Figure 7.9 This figure describes how the membership functions can be defined for each fuzzy set by using percentiles.

Hence, the user must trust the rule-interestingness measures and rule-reduction strategies [9, 10] to get a rule set as reliable as possible. For many applications in bioinformatics, however, some a priori knowledge may be obtained from public information resources. This information may help to give biological significance to the results and, thereby, to support the quality of the rule set.

One of the most important genomic information resources is the Gene Ontology. Thus, many authors have used GO annotations to biologically validate their rule sets. For example, consider one studying associations between entities (genes, proteins, and so on) annotated in the Gene Ontology: *{entityA}* → *{entityB}*, where *entityA* and *entityB* represent any kind of biological entity with annotations in the Gene Ontology (e.g., gene, protein, and so on), or an attribute-value pair containing a biological entity annotated in GO (e.g., *geneA* = *overexpressed*). Ponzoni et al. [52] propose to evaluate the biological significance of the association by analyzing the GO terms in which *entityA* and *entityB* are annotated. By measuring the similarity between the sets of annotations of the items in the antecedent and the consequent of a rule (see Chapter 2 for more information about ontological similarity measures), a value indicating the biological significance of the association can be obtained. High similarity values between the corresponding sets of annotations would help to support the biological significance of the rule.

Since each of the three ontologies describe different biological aspects (i.e., cellular locations, biological processes, and molecular functions), it might be convenient to consider their annotations separately. It is also worth noting that, as the goal is to biologically support the resultant rule set, those unreliable GO annotations should be ignored. For example, it is a common practice to discard those GO annotations with *inferred from electronic annotation* (IEA) evidence code. A scheme of the procedure is shown in Figure 7.10. For rules involving more than one item in the antecedent/consequent, the use of the set operators (e.g., union or intersection) can be investigated to merge the annotations of all the items in the antecedent/consequent.

McIntosh et al. [53] applied a variation of the above strategy. In this case, the authors propose to get a list of statistically over-represented terms in the annotations of the items of the rule. That is, the list of GO terms in which the items of

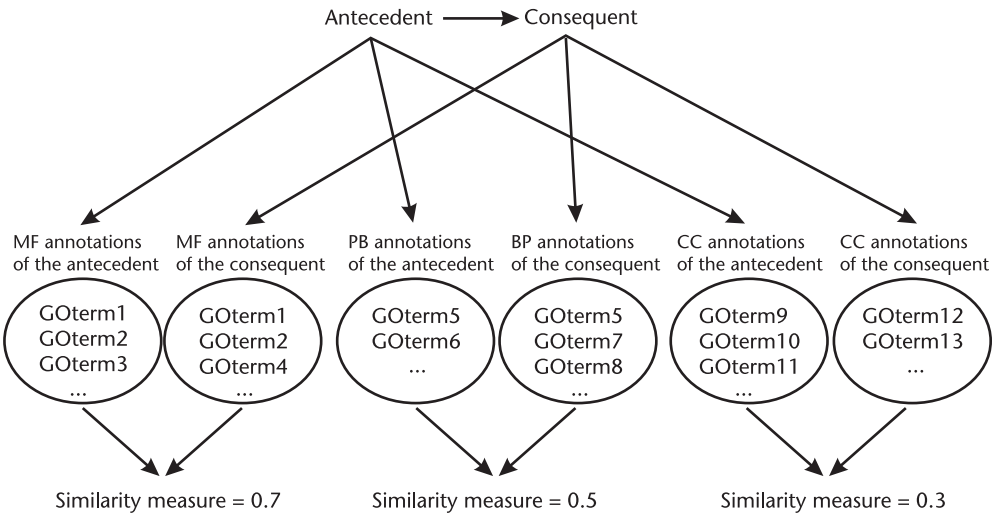


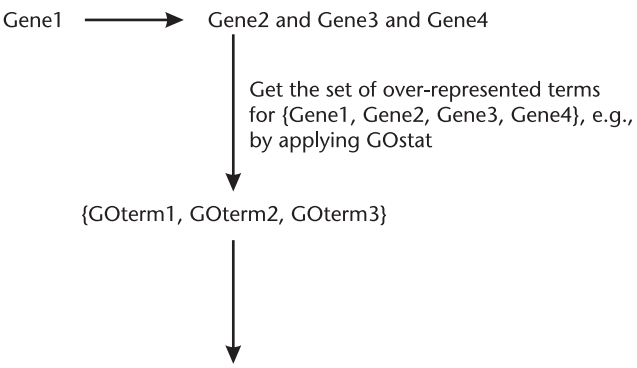
Figure 7.10 An example in which the GO annotations of the antecedent and the consequent of a rule are compared to evaluate the biological significance of the association.

the rule (e.g., genes, proteins, and so on) are annotated is obtained. Statistically over-represented GO terms in this list are identified, and if the items in the antecedent (consequent) share any of these terms with the items in the consequent (antecedent), then the rule may be considered biologically meaningful. The set of statistically over-represented GO terms can be obtained by using some of the existing bioinformatic software packages such as GOstat [54]. Although in the work by McIntosh et al. only rules with one item in the antecedent are considered (Figure 7.11), the proposal can be easily extended to take into account rules with more than one item in the antecedent.

Combinations of the above strategies may also be useful. For example, if rules with several items in the antecedent and the consequent are obtained, one may get the set of over-represented terms in the antecedent and the set of over-represented terms in the consequent. Then, by using an ontological similarity measure (Chapter 4), these two sets can be compared, and a value of their biological relation is obtained.

Another approach using GO similarity measures is that reported by Hoon-Jung et al. [55] to validate their results. In this work, the authors aim to discover conserved domain combinations in S. cerevisae proteins. Thus, they are not interested in the association rules themselves, but in the set of itemsets. Nevertheless, they generate every possible rule from each itemset to calculate the *all-confidence* of the itemset. The *all-confidence* value is the minimum of the confidence values of all the rules that can be generated from an itemset. Hence, it is a measure of the mutual dependency within an itemset.

In this case, each itemset represents a set of protein domains. Their GO annotations are gathered to biologically assess whether these domains might be functionally related. Given an itemset, a set G of GO annotations is obtained, containing the terms annotated to each domain in the combination. All possible GO term pairs are generated from G, and similarity values are calculated for each pair. The



Genes	Over-represented terms	P-value
Gene3, Gene4	GOterm1	0.0000
Gene2	GOterm2	0.0001
Gene1, Gene2, Gene3	GOterm3	0.0020

Figure 7.11 An example in which the rule involving 4 genes is biologically assessed, following the strategy of McIntosh et al. Gene1 shares the term GOterm3 with Gene2 and Gene3. Thereby, it should be considered biologically meaningful.

similarity values are summed and divided by the number of pairs, thus giving a measure of the functional similarity among the domains in the itemset. To calculate the similarity value between two GO terms, the authors make use of FuSSiMeG, which implements the Jiang and Conrath's semantic similarity measure [56]. Obviously, this last approach can be generalized for any other situation in which the interest is on the set of itemsets, and the items contain a biological entity annotated in GO. Figure 7.12 graphically describes the procedure.

Finally, note that the above strategies could be used exactly in the same way to validate fuzzy association rules. That is, the applicability of the above strategies is affected neither by the fuzzy nature of the itemsets that constitute the fuzzy rule, nor by the fuzzy nature of the quality measures (e.g., fuzzy support, fuzzy confidence, and so on) that asses the reliability of the rule.

7.2.3 Other Joint Applications of Association Rules and GO

Association rules and GO have been used together in many other situations in bioinformatics, with very different purposes. This subsection briefly reviews some of these approaches that comprise applications, such as signaling pathways inference, GO annotation prediction, or GO structure analysis.

For example, Bebek et al. [4] integrated gene-expression data, several biological databases (e.g., GO or the Kyoto Encyclopedia of Genes and Genomes [57]), and association rules to infer signaling pathways between two given proteins. The authors first build up a graph in which nodes represent genes, and edges link genes that present correlated expression profiles. Given two proteins, the system looks for every possible path in this graph that connects the corresponding two genes. In order to filter the set of possible paths linking the two genes, the search is guided

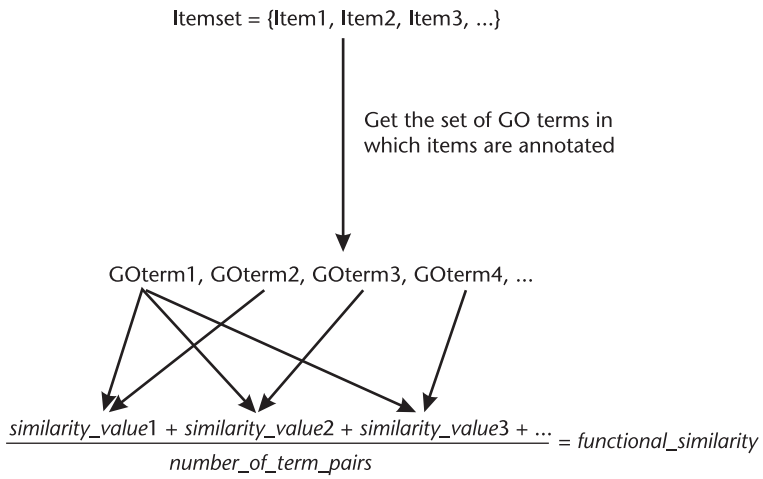


Figure 7.12 An example in which the biological significance of a domain combination is evaluated.

by a set of rules relating GO terms. These rules represent associations between annotations of gene products known to participate in the same pathway.

Association rules have also been used in some cases to automatically generate mappings from different ontologies/databases to GO terms. GO annotation prediction using these mappings may be useful in the manual annotation of genes and gene products. For example, Yu et al. [58] describe *PIPA*, a system for inferring protein functions. The software annotates protein functions by combining the results of multiple programs and databases, such as InterPro, the Conserved Domains Database, and so on. In this case, association rule mining is used to automatically map the different classification schemes of each program/database into GO. Another example of this type is the work by Tveit et al. [59], in which association rules are obtained to find associations between MeSH (the Medical Subject Headings thesaurus, see http://www.nlm.nih.gov/mesh/) and GO terms. In this case, however, another two methodologies are proposed in the same article to obtain the MeSH-to-GO mapping, and they seem to perform better than the association rule mining methodology.

Some authors have studied the GO structure by extracting association rules from GO annotations [60–63]. For example, given the annotations of the genes and gene products of a given species, association rules that represent co-ocurrences and associations among these annotations may unveil implicit associations among GO terms. Furthermore, these associations can be used to find inconsistencies and to infer missing annotations. Moreover, some authors have even proposed a new structure (i.e., a new Gene Ontology layer) capturing biological relations not directly reflected in the present Gene Ontology structure [64].

Finally, on some other occasions, the ultimate objective is not the set of association rules, but the set of itemsets. For example, in the recently developed tool by Carmona-Saez et al. [65], the system aims to obtain statistically over-represented itemsets in the set of annotations of a given group of genes. The annotations are obtained from several sources, such as GO or the Kyoto Encyclopedia of Genes and

Genomes (KEGG) [57]. Klema et al. [66] also try to get sets of interesting itemsets by integrating text mining, functional similarity from GO annotations, and gene-expression data. Further information about these works can be found in the corresponding references.

7.3 Applications for Extracting Knowledge from Microarray Data

Microarray technology makes use of the sequence resources created by the genome projects and other sequencing efforts to monitor the expression of thousands of genes in particular cell samples, times, and conditions. Thus, microarray data provide a global picture of cell activities and open the way to a high-level understanding of its behavior.

The results of a set of microarray experiments are usually presented as a matrix, with as many rows as genes that are being considered and as many columns as experimental conditions that are under study. At this moment, a number of public resources exist that allow one to download and study a wide variety of microarray experiment results [67]. The usual approach to analyzing gene expression datasets consists of applying clustering techniques to obtain groups of correlated genes. This approach has been shown to be very useful, and many clustering techniques have been successfully applied [68–70]. However, gene groupings/clusters can significantly vary, depending on the clustering algorithm used, the similarity measure, and the noise that affects expression data. In addition, the interpretation of resultant clusters is not straightforward, and it usually requires postprocessing work by an expert. Therefore, biological knowledge still needs to be incorporated as a subsequent step to the analysis of gene-expression data.

In this context, association rule mining emerged as an additional tool for analyzing microarray data. This section reviews some of the strategies proposed thus far, focusing on works that jointly use association rules and the Gene Ontology for the analysis of microarray data.

There are two basic approaches to extracting information from microarray data by association rule mining:

1. Obtaining association rules to relate the expression of genes to any type of biological condition/annotation of interest;
2. Obtaining association rules that describe how the expression of one or more genes is associated with the expression of a set of genes.

Obviously the two approaches might be combined to obtain as much information as possible from a given dataset.

Despite the successful applications of association rules for microarray data analysis [2, 3, 71–73], association rule mining for gene-expression data analysis is not without problems. The high number of rules still remains a problem. So far, the solution depends on the use of rule-interestingness measures, rule-reduction techniques [9, 10], and some domain-specific strategies, such as those proposed by Tuzhilin et al. [74]. Moreover, as stated above, obtaining an association rule does

not necessarily mean that a cause-and-effect relationship exists. Obviously, determining the precise nature of an association requires prior biological knowledge and deep investigation.

7.3.1 Association Rules That Relate Gene Expression Patterns with Other Features

As outlined above, association rules can be used to relate the expression of genes to their cellular environment, their functional/structural features, or, in general, to any other type of biological condition/annotation of interest. Several authors followed this idea, and the work by Creighton and Hanash [73] is one of the most popular in this field.

In this type of approach, GO annotations are especially useful in describing molecular functions, biological processes, or cellular locations associated with gene-expression patterns. Several methodologies have been proposed to extract association rules that relate expression data and GO annotations. Carmona-Saez et al. [3] run an association rule mining algorithm over a combined dataset containing the expression data and the GO terms generated, as described in Section 7.2.1. The columns (which form the items) of the data table consist of the set of microarray experiments and the lists of GO terms. The resultant rule set is filtered, so that rules with only one GO annotation in the antecedent and gene-expression level in the consequent are conserved. In addition, a rule $X \rightarrow Y$ is considered redundant, if there is another rule $X' \rightarrow Y'$ with equal or higher values of support, confidence, and improvement (another quality measure [9, 10, 25]), and (1) $X \subset X'$ and $Y \subset Y'$; (2) $X \subset X'$ and $Y = Y'$; or (3) $Y \subset Y'$ and $X = X'$. Redundant rules are also filtered out.

With this methodology, the authors obtain a set of molecular functions, biological processes, or cellular components associated with different gene expression patterns, such as those in Figure 7.13.

Note that a p-value is given for each rule, in addition to the confidence and support values. This value is given by a χ^2-test, under the null hypothesis that the antecedent and the consequent are statistically independent (i.e., the authors make use of a χ^2-test to ensure the correlation of the antecedent and the consequent of the rules).

A similar application is that developed by Martinez et al. [47]. In this work, the authors describe GenMiner, a tool that facilitates the association rule discovery on a data table integrating gene-expression levels, annotations, and any other biological condition. Annotations from several databases are included in the data table: GO annotations, KEGG annotations, bibliographic annotations, and so on.

GenMiner is based on the support-confidence framework, and it implements a version of the Close algorithm [16]. The authors argue that the type of data GenMiner processes is highly correlated and, therefore, that the Apriori algorithm is time and memory-consuming. Moreover, Apriori generates a huge number of rules, many of them redundant. They claim Close is an algorithm specifically designed to deal with these type of data. It limits the search space and reduces the number of dataset scans, thus reducing the execution time and memory usage. Furthermore, it yields a minimal set of rules, thus simplifying the results interpretation.

Antecedent	Consequent											Quality measures		
GO annotation	15m	30m	1h	2h	4h	6h	8h	12h	16 h	20h	24h	Conf.	Supp.	P-value
Cholesterol biosynthesis								−	−	−		38.5	0.1	0.001
Cholesterol biosynthesis						+	+					30.8	0.1	0.001
Angiogenesis						+	+	+				38.5	0.1	0.005
Angiogenesis						+	+	+				30.8	0.1	0.017
Angiogenesis					+	+	+					30.8	0.1	0.014
Chemotaxis					+	+	+	+				12.2	0.1	0.032
Mitosis											+	13.5	0.1	0.147
Positive regulation of cell proliferation						+	+					10.9	0.1	0.158
Blood coagulation					+	+	+					11.1	0.1	0.169
DNA replication									+	+		10.0	0.1	0.235
Cell cycle arrest							−					14.3	0.1	0.257
Mitosis										+	+	10.8	0.1	0.261

Figure 7.13 Some of the rules reported in [3]. The + and – symbols indicate overexpression and underexpression respectively. An empty space indicates that the responding time point does not appear in the rule.

Finally, unlike the work by Carmona-Saez et al. [3], the authors do not impose any rule-template restriction to filter the resultant rule set; they allow every rule to be generated, regardless of the attributes that appear in the antecedent and the consequent, since they argue that every rule yields important information for the biologist.

Another recent work in the field is by Lopez et al. [2]. In this case, the authors combine biclustering techniques, association rules, and GO to extract information from microarrays. It is argued that directly running the association rule mining algorithm over the gene-expression matrix generates a large number of itemsets and rules involving gene-expression levels. They claim that this fact makes the interpretation of the rule set very difficult, since it is hard to identify gene-expression profiles and to relate them with the rest of biological features they consider. Hence, they first run a biclustering algorithm over the expression matrix. Unlike clustering techniques, bicluster methods yield groups (biclusters) of genes that behave similarly under certain conditions (not necessarily all of them), thus avoiding some of the drawbacks of clustering algorithms. Moreover, the biclustering algorithms used in [2] are nonexclusive, thereby they capture the situations in which genes play more than one biological role in conjunction with different groups of genes. In addition, several runs of each biclustering algorithm are carried out with different input parameters to get a broader coverage of the existing gene-expression profiles. Then, a column, containing for each gene the bicluster(s) to which the gene belongs, is included in the data table. Another column is added with the lists of GO terms,

which are selected on the basis of their information content, as explained in Section 7.2.1. The association rule mining algorithm is then run over this data table, and only rules with support, confidence, and certainty factor values [39] greater than certain specified thresholds are generated.

7.3.2 Association Rules to Obtain Relations Between Genes and Their Expression Values

Association rules can also be used so that they describe how the expression of one or more genes is associated with the expression of a set of genes, and, thereby, they are useful in uncovering gene networks. Many works have been developed in this sense, and some efficient algorithms have been specifically designed [71–73].

This strategy looks for associations of the form {*GeneA = expressionA, GeneB = expressionB, …*} → {*GeneC = expression, GeneD = expressionD, …*}, where *expressionA*, *expressionB*, and so on, are discrete expression values that typically represent labels such as *overexpressed*, *underexpressed,* or *not-modified*. On other occasions the expression-level tendency across samples needs to be captured and then *expresionA*, *expressionB*, and so on, represent expression increase or decrease between samples. This can be achieved by simply substituting the original sample values by the differences between samples (see Figure 7.14).

In addition, since rules obtained with this approach relate gene sets, the different strategies proposed in Section 7.2.2 may be used to enhance the interpretability of the results by using GO terms.

The work by Ponzoni et al. [52] can be framed in this type of approach. The authors proposed a machine-learning method based on an optimization procedure to discover regulatory association rules from expression data. They obtain a set of rules of the form

$$\{GeneA = +/-\} \rightarrow \{GeneB +/-\}$$

which may represent one of the following three types of association:

1. *Simultaneous*: the expression level of *GeneB* at time point *i* depends on the expression level of *GeneA* at that time point.
2. *Time delay*: the expression level of *GeneB* at time point *i* depends on the expression level of *GeneA* at time point *i*-1.
3. *Change-based*: when the expression level of *GeneA* changes its state, then the expression level of *GeneB* also changes its state.

One of the main interests of the proposed methodology is the calculation of adaptive regulation thresholds for the discretization of gene-expression values. The authors argue that the gene-expression value required by *geneR* to activate (inhibit) *geneT1* is not necessarily the same value required by the same *geneR* to activate (inhibit) *geneT*2. Hence, they propose a methodology to calculate specific regulation thresholds for each pair of genes.

The biological validation of the results based on the Gene Ontology annotations (see Section 7.2.2) indicates that the gene pairs in the proposed new associations

	GeneA	GeneB	GeneC	...
Time point 1	expraA1	exprB1	expraC1	...
Time point 2	expraA2	expraB2	expraC2	...
Time point 3	expraA3	expraB3	expraC3	...
Time point 4	exp?raA4	expraB4	exprC4	...
...	...	...	...	...
Time point i	exprAi	expraBi	expraCi	...
Time point i + 1	exprAi+1	exprBi+1	exprCi+1	...
...	...	...	...	...

Calculating differences between adjacent samples

	GeneA	GeneB	GeneC	...
Time point 2 – Time point 1	exprA2–expraA1	exprB2–expraB1	exprC2–expraC1	...
Time point 3 – Time point 2	exprA3–expraA2	exprB3–expraB2	exprC3–expraC2	...
Time point 4 – Time point 3	exprA4–expraA3	exprB4–expraB3	exprC4–expraC3	...
...	...	...	...	...
Time point i + 1 – Time point i	exprAi+1–expraAi	exprBi+1–expraBi	exprCi+1–expraCi	...
...	...	...	...	...

Figure 7.14 An example in which the differences between adjacent samples are calculated so that the rules can capture associations between expression-level tendencies.

seem to be functionally related, according to GO, thereby supporting the hypothesis that these gene pairs may be regulatory related.

A similar work is the one by McIntosh et al. [53]. In this case, the authors propose an efficient algorithm to mine association rules from gene-expression data. The algorithm makes use of a tree structure that allows the avoidance of any support constraint, since it is able to prune the search space by estimating the confidence of the rules that are about to be generated. In this case, they only consider rules of the form *{GeneA =expressed/not-expressed}* → *{GeneB = expressed/not-expressed, GeneC = expressed/not-expressed, GeneD = expressed/not-expressed* …}, that is, rules with only one item (gene) in the antecedent and several items (genes) in the consequent. Gene Ontology annotations are also used in this case to validate the rule set, (see Section 7.2.2 for further details).

To finalize, it is worthy to note the lack of works in the literature that make use of fuzzy association rules to analyze microarray data and, in general, any other type of biological information. In addition to all the previously mentioned advantages that fuzzy sets have over classical crisp sets (see Section 7.1.4), fuzzy sets are known to perform better when dealing with imprecise and noisy data. Microarrays and, in general, any kind of biological information source, are likely to be imprecise and quite noisy.

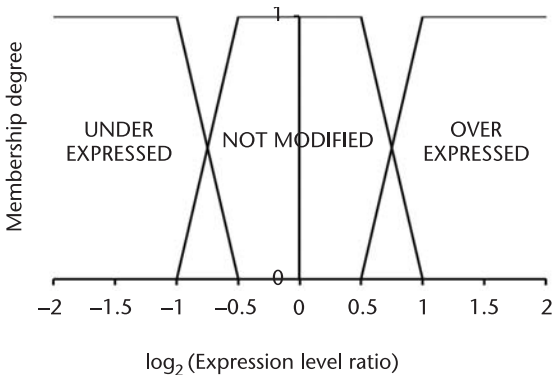


Figure 7.15 The figure shows a straightforward way of defining the fuzzy concepts *overexpressed, underexpressed* and *not-modified.*

A straightforward definition of the fuzzy concepts *overexpressed* and *underexpressed* could be, for example, that of Figure 7.15, taking into account that it is usually accepted that genes that express more than one fold with respect to the control sample are overexpressed, and genes that express less than one fold with respect to the control sample are underexpressed. By using algorithms such as those reviewed in Section 7.1.4 or by adapting any of those specifically designed for microarray analysis, such as those cited in Section 7.3, fuzzy association analysis can be carried out with not much more effort than a crisp analysis.

Moreover, fuzzy rules are easy to understand, since they are very similar to the way a person might express knowledge. This makes them especially suitable for their application in this field in which experts must validate the results. Therefore, there is much room for improvement regarding the development and application of fuzzy techniques, and particularly, fuzzy association rule mining techniques for treating biological information.

Acknowledgements

This work has been carried out as part of projects P08-TIC-4299 of J. A., Sevilla and TIN2009-13489 of DGICT, Madrid.

References

[1] Kanehisa, M., and P. Bork, "Bioinformatics in the Post-Sequence Era," *Nature Genet*, Vol. 33, 2003, pp. 305–310.

[2] Lopez, F. J., et al., "Fuzzy Association Rules for Biological Data Analysis: A Case Study on Yeast," *BMC Bioinformatics*, Vol 9, 2008, p. 107.

[3] Carmona-Saez, P., et al., "Integrated Analysis of Gene Expression by Association Rules Discovery," *BMC Bioinformatics*, Vol. 7, 2006, p. 54.

[4] Bebek, G., and J. Yang, "PathFinder: Mining Signal Transduction Pathway Segments from Protein-Protein Interaction Networks," *BMC Bioinformatics*, Vol. 8, 2007, pp. 335–347.

[5] Morgan X. C., et al., "Predicting Combinatorial Binding of Transcription Factors to Regulatory Elements in the Human Genome by Association Rule Mining," *BMC Bioinformatics*, Vol. 8, 2007, p. 445.

[6] The Gene Ontology Consortium, "Gene Ontology: Tool for the Unification of Biology," *Nature Genet.*, Vol. 25, 2000, pp. 25–29.

[7] Agrawal, R., T. Imielinski, and A. Swami, "Mining Association Rules Between Sets of Items in Large Databases", *Proc. of the ACM SIGMOD INTL Conf. on Management of Data (ACM SIGMOD 93)*, Washington, D.C., 1993, pp. 207–216.

[8] Freitas, A., "Are We Really Discovering 'Interesting' Knowledge from Data?" *Expert Update (the BCS-SGAI Magazine)*, Vol. 9, No. 1, 2006, pp. 41–47.

[9] Geng, L., and H. J. Hamilton, "Interestingness Measures for Data Mining: A Survey,"*ACM Computing Surveys*, Vol. 38, No. 3, Article 9, 2006, pp. 1–32.

[10] Ceglar, A., and J. F. Roddick, "Association Mining," *ACM Computing Surveys*, Vol. 38, No. 2, Article 5, 2006, pp. 1–42.

[11] Goethals, B., and M. J. Zaki, "Advances in Frequent Itemset Mining Implementations: Report on FIMI'03," *SIGKDD Explorations*, Vol. 6, No. 1, pp. 109–117.

[12] Zaki, M., et al., "New Algorithms for Fast Discovery of Association Rules," *Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining (KDD'97)*, Menlo Park, California, August 14–17, 1997, pp. 283–296.

[13] Houtsma, M., and A. Swami, "Set Oriented Mining of Association Rules," Technical Report *RJ 9567*, IBM Almaden Research Center, 1993.

[14] Han, J., and J. Pei, "Mining Frequent Patterns by Pattern Growth: Methodology and Implications," *SIGKDD Explorations*, Vol. 2, No. 2, 2000, pp. 14–20.

[15] Pei, J., J. Han, and L. V. S. Lakshamanan, "Mining Frequent Itemsets with Convertible Constraints," *Proc. of the 17th Int. Conf. on Data Engineering (ICDE'01)*, Heidelberg, Germany, April 2–6, 2001, pp. 433–442

[16] Pasquier, N., et al., "Efficient Mining of Association Rules Using Closed Itemset Lattices," *Informa. Syst.*, 1999, Vol. 24, No. 1, pp. 25–46.

[17] Bastide, Y., et al., "Mining Frequent Patterns with Counting Inference," *SIGKDD Explorations*, Vol. 2, No. 2, 2000, pp. 66–75.

[18] Calders, T., and B. Goethals, "Mining All Non-Derivable Frequent Itemsets," *Lecture Notes in Computer Science*, Vol. 2,431, 2002, pp. 74–85.

[19] Bykowski, A., and C. Rigotti, "A Condensed Representation to Find Frequent Patterns," *Proc. of the 20th ACM SIGMOD-SIGACT-SIGART Symp. on the Principles of Database Systems*, Santa Barbara, California, May 21–23, 2001, pp. 267–273.

[20] Srikant, R., and R. Agrawal, "Mining Generalized Association Rules," *Proc. of the 21st Conf. on Very Large Databases (VLDB'95)*, Zurich, Switzerland, September 11–15, 1995, pp. 407–419.

[21] Han, J., and Y. Fu, "Discovery of Multiple-Level Association Rules from Large Databases," *Proc. of the 21st Conf. on Very Large Databases (VLDB'95)*, Zurich, Switzerland, September 11–15, 1995, pp. 420–431.

[22] Hipp, J., et al., "A New Algorithm for Faster Mining of Generalized Association Rules," *Proc. of the 2nd European Symp. on Principles of Data Mining and Knowledge Discovery (PKDD'98)*, Nantes, France, September 23–26, 1998, pp. 74–82.

[23] Borgelt, C., "Efficient Implementations of Apriori and Eclat," *Proc. 1st IEEE ICDM Workshop on Frequent Item Set Mining Implementations (FIMI 2003)*, Melbourne, FL, November 19, 2003.

[24] Bodon, F., "A Fast Apriori Implementation," *Proc. of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, Melbourne, FL, November 19, 2003.

[25] Tan, P. N., M. Steinbach, and V. Kumar, *An Introduction to Data Mining*, Boston, MA: Addison-Wesley Longman Publishing; 2005, p. 769.

[26] Hipp, J., U. Güntzer, and G. Nakhaeizadeh, "Algorithms for Association Rule Mining—A General Survey and Comparison," *SIGKDD Explorations*, Vol. 2, No. 1, 2000, pp. 58–65.

[27] Srikant, R., and R. Agrawal, "Mining Quantitative Association Rules in Large Relational Databases," *Proc. of the ACM SIGMOD (SIGMOD'96)*, Montreal, Canada, June 4–6, 1996, pp. 1–12.

[28] Miller, R. J., and Y. Yang, "Association Rules over Interval Data," *Proc. of the ACM SIGMOD (SIGMOD'97)*, Tucson, Arizona, May 13–15, 1997, pp. 452–461.

[29] Chen, G., Q. Wei, and E. E. Kerre, "Fuzzy Logic in Discovering Association Rules: An Overview," *Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques*, Heidelberg, Germany: Springer, 2006, pp. 459–493.

[30] Delgado, M., et al., "Mining Fuzzy Association Rules: An Overview," *Proc. of the BISC Int. Workshop on Soft Computing for Internet and Bioinformatics*, Berkeley, CA, Decmber 15–19, 2003.

[31] Chien, B. C., Z. L. Lin, and T. P. Hong, "An Efficient Clustering Algorithm for Mining Fuzzy Quantitative Association Rules," *Proc. of the 9th Int. Fuzzy Systems Assoc. World Congress*, Vancouver, Canada, July 25–28, 2001, pp. 1306–1311.

[32] Gyenesei, A., "A Fuzzy Approach for Mining Quantitative Association Rules," *TUCS Technical Report 336*, Department of Computer Science, University of Turku, Finland, 2000.

[33] Fu, A., et al., "Finding Fuzzy Sets for the Mining of Fuzzy Association Rules for Numerical Attributes," *Proc. of the 1st Int. Symp. on Intelligent Data Engineering and Learning (IDEAL'98)*, Hong Kong, October 15–17, 1998, pp. 263–268.

[34] Alcala, R., et al., "Genetic Learning of Membership Functions for Mining Fuzzy Association Rules," *Proc. of the Fuzzy Systems Conf. (FUZZ-IEEE 2007)*, London, U.K., July 23–26, 2007, pp.1538–1543.

[35] Dubois, D., H. Prade, and T. Sudkamp, "A Discussion of Indices for the Evaluation of Fuzzy Associations in Relational Databases," *Proc. of the 10th Int. Fuzzy Systems Association World Congress (IFSA-03)*, Istanbul, Turkey, June 29–July 2, 2003, pp. 111–118.

[36] Dumitrescu, D., B. Lazzerini, and L. C. Jain, *Fuzzy Sets and Their Application to Clustering and Training*, New York: CRC Press, 2000, p. 622.

[37] Dubois, E., E. Hüllermeier, and H. Prade, "A Systematic Approach to the Assessment of Fuzzy Association Rules," *Data Mining and Knowledge Discovery*, Vol. 2, 2006, pp. 167–192.

[38] Glass, D. H., "Fuzzy Confirmation Measures," *Fuzzy Sets and Systems*, Vol. 159, 2008, pp. 475–490.

[39] Delgado, M., et al., "Fuzzy Association Rules: General Model and Applications," *IEEE Trans. Fuzzy Syst.*, Vol. 11, No. 2, 2003, pp. 214–225.

[40] Lee, J. H., and H. L. Kwang, "An Extension of Association Rules Using Fuzzy Sets," *Proc. of the 7th Int. Fuzzy Systems Assoc. World Congress*, Prague, Czech Republic, June 25–29, 1997, pp. 399–402.

[41] Au, W. H., and K. C. C. Chan, "An Effective Algorithm for Discovering Fuzzy Rules in Relational Databases," *Proc. of the IEEE Int. Conf. on Fuzzy Systems*, Anchorage, Alaska, July 25–29, Vol. 2, 1998, pp. 1314–1319.

[42] Au, W. H., and K. C. C. Chan, "FARM: A Data Mining System for Discovering Fuzzy Association Rules," *Proc. of the FUZZ-IEEE'99*, Seoul, South Korea, August 22–25, Vol. 3, 1999, pp.22–25.

[43] Zhang, W., "Mining Fuzzy Quantitative Association Rules," *Proc. of the 11th Int. Conf. on Tools with A.I.*, Chicago, Illinois, November 8–10, 1999, pp. 99–102.

[44] Hen, T. P., C. S. Kuo, and S. C. Chi, "A Fuzzy Data Mining Algorithm for Quantitative Values," *Proc. of the 3rd Int. Conf. on Knowledge-Based Intelligent Information Engineering Systems*, Adelaide, Australia, August 31–September 1, 1999, pp. 480–483.

[45] Chen, G., Q. Wei, and E. E. Kerre, "Fuzzy Data Mining: Discovery of Fuzzy Generalized Association Rules," *Recent Issues on Fuzzy Databases*, New York: Physica-Verlag (Springer), 2000, pp. 45–66.

[46] Hong, T.P., K. Y. Ling, and S. L. Wang, "Fuzzy Data Mining for Interesting Generalized Association Rules," *Fuzzy Sets and Systems*, Vol. 138, 2003, pp. 255–269.

[47] Martinez, R., C. Pasquier, and N. Pasquier, "GenMiner: Mining Informative Association Rules from Genomic Data," *Proc. of the IEEE Int. Conf. on Bioinformatics and Biomedicine*, Silicon Valley, California, November 2–4, 2007, pp. 15–22.

[48] Al-Shahrour, F., R. Diaz-Uriarte, and J. Dopazo, "FatiGO: a Web Tool for Finding Significant Associations of Gene Ontology Terms with Groups of Genes," *Bioinformatics*, Vol. 20, No. 4, 2004, pp. 578–580.

[49] Mateos, A., et al., "Supervised Neural Networks for Clustering Conditions in DNA Array Data after Reducing Noise by Clustering Gene Expression Profiles," *Methods of Microarray Data Analysis II*, Boston, MA: Kluwer Academics Publishers, 2002, pp. 91–103.

[50] Alterovitz, G., et al., "GO PaD: The Gene Ontology Partition Database," *Nucleic Acids Research*, Vol. 35, 2007, pp. D322–D327.

[51] Wang, K., et al., "Top Down FP-Growth for Association Rule Mining," *Proc. of the 6th Pacific Area Conf. on Knowledge Discovery and Data Mining*, Taipei, Taiwan, May 6–8, 2002, pp. 334–340.

[52] Ponzoni, I., et al., "Inferring Adaptive Regulation Thresholds and Associaion Rules from Gene Expression Data Through Combinatorial Optimization Learning," *IEEE Trans. on Computational Biology and Bioinformatics*, Vol. 4, No. 4, 2007, pp. 624–634.

[53] McIntosh, T., and S. Chawla, "High-Confidence Rule Mining for Microarray Analysis," *IEEE Trans. on Computational Biology and Bioinformatics*, Vol. 4, No. 4, 2007, pp. 611–623.

[54] Beissbarth, T., and T. Speed, "GOstat: Find Statistically Over-Represented Gene Ontologies Within Gene Groups," *Bioinformatics*, Vol. 20, No. 9, 2004, pp. 1464–1465.

[55] Jung, S. H., et al., "Identification of Conserved Domain Combinations in S. Cerevisae Proteins," *Proc. of the 7th IEEE Int. Conf. on Bioinformatics and Bioengineering (BIBE 2007)*, Boston, Massachusetts, October 14–17, 2007, pp. 14–20.

[56] Couto, F., M. Silva, and P. Coutinho, "Implementation of a Functional Semantic Similarity Measure Between Gene-Products," Technical Report, Dept. of Informatics, Faculty of Sciences, Univ. of Lisbon, 2003.

[57] Kanehisa, M., and S. Goto, "KEGG: Kyoto Encyclopedia of Genes and Genomes," *Nucleic Acids Research*, Vol. 28, 2000, pp. 27–30.

[58] Yu, C., et al., "The Development of PIPA: An Integrated and Automated Pipeline for Genome-Wide Protein Function Annotation," *BMC Bioinformatics*, Vol. 9, 2008, pp. 52–62.

[59] Tveit, H., T. Mollestad, and A. Lægreid, "The Alignment of the Medical Subject Headings to the Gene Ontology and Its Application in Gene Annotation," *Proc. of the 4th Int. Conf. on Rough Sets and Current Trends in Computing (RSCTC 2004)*, Uppsala, Sweden, June 1–5, 2004, pp. 798–804

[60] Burgun, A., et al., "Dependence Relations in Gene Ontology: A Preliminary Study," *Proc. of the Workshop on the Formal Architecture of the Gene Ontology*, Leipzig, Germany, **May 28–29**, 2004.

[61] Kumar, A., B. Smith, and C. Borgelt, "Dependence Relationships Between Gene Ontology Terms Based on TIGR Gene Product Annotations," *Proc. of the 3rd Int.Workshop on Computational Terminology*, Geneva, Switzerland, August 29, 2004, pp. 31–38.

[62] Bodenreider, O., "Non-Lexical Approaches to Identifying Associative Relations in the Gene Ontology," *Proc. of the Pacific Symp. on Biocomputing 2005 (PSB 2005)*, Lihue, HI, January 4–8, 2005, pp. 91–102.

[63] Burgun, A., and O. Bodenreider, "An Ontology of Chemical Entities Helps Identify Dependence Relations Among Gene Ontology Terms," *Proc. of the 1st Int. Symp. on Semantic Mining in Biomedicine (SMBM 2005)*, Hinxton, U.K., April 10–13, 2005.

[64] Myhre, S., et al., "Additional Gene Ontology Structure for Improved Biological Reasoning," *Bioinformatics*, Vol. 22, No. 16, 2006, pp. 2020–2027.

[65] Carmona-Saez, P., "GENECODIS: A Web-Based Tool for Finding Significant Concurrent Annotations in Gene Lists," *Genome Biology*, Vol. 8, 2007, p. R3.

[66] Klema, J., "Constraint-Based Knowledge Discovery from SAGE Data," *Silico Biology*, Vol. 8, 2008, pp. 157–175.

[67] Gardiner-Garden, M., and T. G. Littlejohn, "A Comparison of Microarray Databases," *Briefings in Bioinformatics*, Vol. 2, No. 2, 2001, pp. 143–158.

[68] Eisen, M. B., et al., "Cluster Analysis and Display of Genome-Wide Expression Patterns," *Proc.of the National Academy of Sciences*, 1998, Vol. 95, pp. 14863–14868.

[69] Tavazoie, S., et al., "Systematic Determination of Genetic Network Architecture," *Nature Genet.*, Vol. 22, 1999, pp. 281–285.

[70] Liotta, L., and E. Petricoin, "Molecular Profiling of Human Cancer," *Nature Reviews Genetics*, Vol. 1, 2000, pp. 48–56.

[71] Huang, Z., et al., "Large-Scale Regulatory Network Analysis from Microarray Data: Modified Bayesian Network Learning and Association Rule Mining," *Decision Support Systems*, 2007, Vol. 43, 2007, pp. 1207–1225.

[72] Jiang, X. R., and L. Grenwald, "Microarray Gene Expression Data Association Rules Mining Based on bsc-tree and fis-tree," *Data and Knowledge Engineering*, Vol. 53, No. 1, 2005, pp. 3–29.

[73] Georgii, E., et al., "Analyzing Microarray Data Using Quantitative Association Rules," *Bioinformatics*, Vol. 21, 2005, pp. ii123–ii129.

[74] Creighton, C., and S. Hanash, "Mining Gene Expression Databases for Association Rules," *Bioinformatics*, Vol. 19, No. 1, 2003, pp. 79–86.

[75] Tuzhilin, A., and G. Adomavicius, "Handling Very Large Numbers of Association Rules in the Analysis of Microarray Data," *Proc. of the 8th Int. Conf. on Knowledge Discovery and Data Mining (KDD'02)*, Edmonton, Canada, July 23–26, 2002, pp. 396–404.

CHAPTER 8

Text Summarization Using Ontologies

Henrik Bulskov and Troels Andreasen

A summary is a comprehensive description that grasps the essence of a subject. A text, a collection of text documents, or a query answer can be summarized by simple means, such as an automatically generated list of the most frequent words or advanced by a meaningful textual description of the subject. Between these two extremes are summaries by means of selected concepts exploiting background knowledge providing selected key concepts. We address in this chapter an approach where conceptual summaries are provided through a conceptualization as given by an ontology. The idea is to restrict a background ontology to the set of concepts that appears in the text to be summarized and thereby provide a structure, a so-called instantiated ontology, that is specific to the domain of the text and can be used to condense to a summary, not only quantitatively but also conceptually covering the subject of the text. The problem of how to derive ontologies from resources, such as lexicons, is considered, with focus on a general, as well as the biomedical domain.

8.1 Introduction

The purpose of a summary is to provide a simplification to highlight the major points from the subject (e.g., a text or a set of texts, such as a query answer). The aim is to provide a summary that grasps the essense of the subject.

Most common are summaries, as those provided manually by readers or authors as a result of intellectual interpretation. However, summeries can also be provided automatically. One approach, in the question answering style, such as this is investigated in, for instance, the DUC and TREC conferences [5–7], is to provide a full natural-language generation, based summary construction, while a less ambiguous, in the same tradition, is rather to perform a sentence selection from the text to be summarized.

In the other end, the most simple approach is to select a reasonably short list of words, among the most frequent and/or the most characteristic words from the set of words found in the text to be summarized. So, rather than a coherent text, the summary is simple a set of items.

Summaries in the approach presented here are also sets of items, but they involve improvements over the simple set-of-words approach in two respects. First,

we go beyond the level of keywords and aim to provide conceptual descriptions from concepts identified and extracted from the text. Second, we involve background knowledge in the form of an ontology. Strictly, these two aspects are closely related—to use the conceptualization in the ontology, we need means to map from words and phrases in the text to concepts in the ontology.

Summarization is a process of transforming sets of similar low-level objects into more abstract conceptual representations [19], and more specifically, a summary for a set of concepts is an easy to grasp and short description—in the form of a smaller set of concepts. For instance {*car*, *house*} as a summary for {*convertible*, *van*, *cottage*, *estate*} or {*dog*} as a summary for {*poodle*, *alsatian*, *golden retriever*, *bulldog*}.

In this chapter we present two different directions to conceptual summaries as answers to queries. In both cases, an ontology plays a key role as a reference for the conceptualization. The general idea is from a world knowledge ontology to form a so-called *instantiated ontology* by restricting it to a set of instantiated concepts.

First, we consider a strictly ontology-based approach in which summaries are derived solely from the instantiated ontology. Second, we consider conceptual clustering over the instantiated concepts based on a semantic similarity measure such as the shortest path [12]. The semantic grouping that results from the clustering process is then summarized, using either the least upper bounds of the clusters or by the introduction of fuzzy least upper bounds. The advantage of using the latter is that they enable summaries that are more accurate and more tolerant with regard to noise in clusters.

The approach presented here can be seen as an approach to conceptual querying, in which a set of concepts can be examined.

The general idea, in the approach presented here, is to restrict a general world-knowledge ontology to the given set of concepts, extending this with relations and related concepts and, thereby, providing a structure for navigation and further investigation of the concepts. A conceptual investigation of a set of documents can be performed by extracting the set of concepts appearing in the documents and by providing a means for navigation and retrieval within the set of extracted concepts.

This chapter is organized as follows. First, we introduce the general ontology, extraction of conceptual descriptions, and the instantiated ontology. Second, we describe the various approaches to conceptual summaries, with a special focus on the concept of fuzzy least upper bound. Third, the approaches are illustrated using WordNet [11] and SemCor [10]. Finally we present a conclusion and give some pointers to future research.

8.2 Representing Background Knowledge—Ontology

Background knowledge is knowledge that complements the primary target data (the text or text collection or database) that is the subject of the summarization with information that is essential to the understanding of this. Background knowledge can take different forms, varying from simple lists of words to formal representations. To provide, in the question answering style, a full natural language generation-

based summary, a means for reasoning within the domain, as well as a means for processing language expressions is needed. Therefore, background knowledge should include axiomatic formalization of essential domain knowledge, as well as knowledge to guide the natural-language synthesis process. In this context, however, our goal is conceptual summaries provided as sets of words or concepts, so background knowledge to support this can range from unstructured lists of words to ontologies.

A simple list of words can be applied as a filter, mapping from a text to the subset of the word list that appears in the text. Such a controlled list of keywords or vocabulary of topics can, by obvious means, be improved to also capture morphology by stemming or inflection patterns. For summary purposes, however, we will have to rely on such course-grained principles as statistics on frequencies to reduce the number of items of a list or to obtain an easy-to-grasp summary. What is needed to obtain significant improvement is a structure that relates individual words and thereby supports fusion into commonly related items in the contraction toward sufficiently brief summaries. In addition to this, the presence of relations introduces the element of definition by related items and thus justifies the notion as a structure of concepts rather than a list of words. So taxonomies, partonomies, semantic networks and ontologies are structures that potentially contribute also to knowledge-based summarization. Our main focus here is on ontologies ordered around taxonomic relationship. Rather than the common description-logic-based approach we choose here a simpler concept, algebraic approach to ontologies.

One important rationale for this is that our goal here is not ontological reasoning in general, but rather extraction of sets of mapped concepts and manipulation of such sets (e.g., contraction). Another reason is that the concept algebraic approach has an inherent and very significant notion of generativity, where the ontology also includes compound concepts that can be formed by means of other concepts.

8.2.1 An Algebraic Approach to Ontologies

Let us consider a basis taxonomy that situates a set of atomic term concepts A in a multiple-inheritance hierarchy. Based on this, we define a generative ontology by generalization of the hierarchy to a lattice and by introducing a (lattice-algebraic) concept language (description language) that defines an extended set of well-formed concepts, including both atomic and compound term concepts.

The concept language used here, ONTOLOG [9], has, as basic elements, concepts and binary relations between concepts. The algebra introduces two closed operations sum and product on concept expressions φ and ψ, where $(\varphi + \psi)$ denotes the concept, being either φ or ψ, and $(\varphi \times \psi)$ denotes the concept being φ and ψ (also called *join* and *meet*, respectively).

Relationships r are introduced algebraically, by means of a binary operator (:), known as the Peirce product $(r : \varphi)$, which combines a relation r with an expression φ. The Peirce product is used as a factor in conceptual products, as in $x \times (r{:}y)$, which can be rewritten to form the feature structure $x[r{:}\mathrm{y}]$, where $[r : y]$ is an attribution of the concept x. Thus, we can form compound concepts by attri bution.

Given a set of atomic concepts A and semantic relations R, the set of well-formed terms L is

$$L = \{A\} \cup \left\{ x\left[r_1 : y_1, \cdots, r_n : y_n\right] \middle| x \in A, r_i \in R, y_i \in L \right\} \tag{8.1}$$

Compound concepts can thus have multiple as well as nested attributions. For instance, with R = {WRT, CHR, CBY, TMP, LOC,...}[1] and A = {*entity*, *physical entity*, *abstract entity*, *location*, *town*, *cathedral*, *old*} we get:

$$\begin{aligned} L = & \\ & \{ entity, physical\_entity, abstract\_entity, \\ & location, town, cathedral, old, \\ & \cdots, cathedral\left[\mathrm{LOC} : town, \mathrm{CHR} : old\right], \\ & cathedral\left[\mathrm{LOC} : town\left[\mathrm{CHR} : old\right]\right], \ldots \} \end{aligned}$$

8.2.2 Modeling Ontologies

Obviously modeling ontologies from scratch is the best way to ensure that the result will be correct and consistent. However, for many applications the effort it takes is simply not at disposal and manual modeling has to be restricted to narrow and specific subdomains, while the major part have to be derived from relevant sources. Sources that may contribute to the modeling of ontologies may have various forms. A taxonomy is an obvious choice, and it may be supplemented with, for instance, word and term lists as well as dictionaries for the definition of vocabularies and for the handling of morphology. Among the obviously useful resources are the Semantic Network WordNet [11] and the Unified Medical Language System (UMLS) [4] and several other resources in the biomedical science area.

To go from a resource to an ontology is not necessarily straightforward, but if the goal is a generative ontology, and the given resource is a taxonomy, one option is to proceed as follows. Given a taxonomy T over the set of atomic concepts A and a language L, over A for a given set of relations R, being derived as indicated in (8.1). Let $\hat{T}$ be the transitive closure of T. $\hat{T}$ can be generalized to an inclusion relation ≤ over all well-formed terms of the language L by the following:

$$\begin{aligned} \leq = & \hat{T} \\ & \cup \left\{ \left\langle x\left[\ldots, r : z\right], y\left[\cdots\right] \right\rangle \middle| \left\langle x\left[\ldots\right], y\left[\cdots\right] \right\rangle \in \hat{T} \right\} \\ & \cup \left\{ \left\langle x\left[\ldots, r : z\right], y\left[\cdots, r : z\right] \right\rangle \middle| \left\langle x\left[\ldots\right], y\left[\cdots\right] \right\rangle \in \hat{T} \right\} \\ & \cup \left\{ \left\langle z\left[\ldots, r : x\right], z\left[\cdots, r : y\right] \right\rangle \middle| \left\langle x, y \right\rangle \in \hat{T} \right\} \end{aligned} \tag{8.2}$$

1. For *with respect to, characterized by, caused by, temporal, location,* respectively.

where repeated ... denote zero or more attributes of the form $r_i : w_i$.

The general ontology $O = (L, \leq, R)$ thus encompasses a set of well-formed expressions L, derived in the concept language from a set of atomic concepts A, an inclusion relation generalized from the taxonomy relation in T, and a supplementary set of semantic relations R. For $r \varepsilon R$, we obviously have $x[r : y] \leq x$, and that $x[r : y]$ is in relation r to y. Observe that O is generative and that L therefore is potentially infinite.

An example is given in Figure 8.2 showing a segment of a generative, ontology built with WordNet as a resource.

8.2.3 Deriving Similarity

An ontology that covers a document collection may provide an excellent means to survey and give perspective to the collection, however, as far as access to documents is concerned, ontology reasoning is not the most obvious evaluation strategy, as it may well entail scaling problems. Applying measures of similarity derived from the ontology is a way to replace reasoning with simple computation still influenced by the ontology.

One obvious way to measure similarity in ontologies, given the graphical representation, is to evaluate the distance between the concepts being compared, where a shorter distance implies higher similarity and vice versa.

A number of different ontological-similarity measures along this line have been proposed over the years. *Shortest Path Length* [12] forms the basis of a group of measures classified as path length approaches. The *Weighted Shortest Path* [15] is a generalization of *Shortest Path Length,* in which weights are assigned to relations in the ontology. Two different alternatives are *Information Content* [16] and *Weighted Shared Nodes* [17], where the former uses the probability of encountering concepts in a corpus to define the similarity between concepts, and the latter uses the density of concepts shared by the concepts being compared to measure the similarity.

8.3 Referencing the Background Knowledge—Providing Descriptions

As already indicated, the approach involves surveying text through the ontology provided and delivering summaries on top of the conceptualization of the ontology. For this purpose, we need to provide a description of the text to be summarized in terms of the concepts in the ontology. So words and/or phrases must be extracted from the text and mapped into the ontology. This is a knowledge extraction problem, and obviously such knowledge extraction can span from full, deep, natural -language processing (NLP) to simplified shallow processing methods.

Here we will consider the latter, due to the counterbalance between the need for a full interpretation and the computational complexity of getting it. A very simple solution would match words in text with labels of concepts in the ontology, and hence, make a many-to-many relation between words in text and labels in the ontology that just accepts the ambiguity of natural language. Improvements can

easily be obtained through pattern-based information extraction/text mining and through methods in natural-language processing.

First, a heuristic part of speech tagging can be performed on the text, and provided that word classes are assigned to the concepts given in the ontology, this enables a word-class-based disambiguation.

Second, a stemming or, provided lexical information is available, a transformation to a standardized inflectional form can significantly improve the matching.

Third, given part-of-speech tagged input, simple syntactic natural language grammars can be used to chunk words together, forming utterances or phrases [3], that can be used as the basis for matching against compound concepts in the ontology. Obviously, the matching of chunks from the text and concepts in the ontology is, in principle, the same complex NLP problem over again, but the chunks identified will often correspond to meaningful concepts and, therefore, lead to a more refined and better result of the matching and, in addition, allow for simple pattern-based approaches. We refer to [18] and [1] for more refined approaches. Here we will cover only a simple pattern-based approach.

Finally, some kind of word sense disambiguation [20] can be introduced in order to narrow down the possible readings of words, hence, ideally mapping words of phrases to exactly one concept in the ontology.

A very simple approach along these lines is the following. Given a part-of-speech-tagged and NP-chunked input, a grammar for interpretation of the chunks is the following:

$$\begin{aligned} &\text{Head} ::= N \\ &NP ::= A^{*}\, N^{*}\ \text{Head} \mid NP\ P\ NP \end{aligned} \tag{8.3}$$

where A, N, and P are placeholders for adjective, noun, and preposition, respectively. A very course-grained mapping strategy on top of this interpretation can be formed using the following transition rules, in which premodifying adjectives relates to the head through *characterized by* (CHR) while premodifying composite nouns and prepositions both relate through *with respect to* (WRT):

$$\begin{aligned} &A_1,\cdots,A_n N_1,\cdots,N_m Head \rightarrow \\ &\quad Head\left[CHR:A_1,\cdots,CHR:A_n,WRT:N_1,\cdots,WRT:N_m\right] \\ &NP\left(P\ NP\right)_1,\cdots,\left(P\ NP\right)_n \rightarrow \\ &\quad NP\left[WRT:NP_1,\cdots,WRT:NP_n\right] \end{aligned} \tag{8.4}$$

To test this approach, we consider the the metathesaurus in the Unified Medical Language System (UMLS) [13] as a resource and build a generative ontology from this. For part-of-speech tagging and phrase chunking we use the MetaMap application [2].

Consider the following utterance[2] as an example:

> [...] the plasma patterns of estrogen and progesterone under gonadotropic stimulation simulating early pregnancy [...]

The first part of the analysis leads to part of speech tagging and phrase recognition as follows:

Phrase	*Type*	*Word*	*POS*
Noun phrase	Det	The	Det
	Mod	Plasma	Noun
	Head	Patterns	Noun
Preposition	Prep	Of	Prep
	Head	Estrogen	Noun
	Conj	And	Conj
Noun phrase	Head	Progesterone	Noun
Preposition	Prep	Under	Prep
	Mod	Gonadotropic	Adj
	Head	Stimulation	Noun
	Verb	Simulating	Verb
Noun phrase	Mod	Early	Adj
	Head	Pregnancy	Noun

By applying the grammar (3), this can be transformed into the following three noun phrases:

plasma/N patterns/N of/P estrogen/N
progesterone/N under/P gonadotropic/A stimulation/N
early/A pregnancy/N

and by using the transition rules (4), we can produce the following compound expressions:

patterns[WRT: plasma, WRT: estrogen]
progesterone[WRT:stimulation[chr:gonadotropic]]
pregnancy[CHR:early]

Then we can attach the mapping from words in these expressions to node identifiers in the Metathesaurus given by MetaMap:

patterns{C0449774}
  [WRT: plasma{C0032105, C1546740}, WRT: estrogen{C0014939}]
progesterone{C0033308}
  [WRT:stimulation{C1948023,C1292856}[chr:gonadotropic{C1708248}]]
pregnancy{C0425965,C0032961}
  [CHR:early{C1279919}]

Naturally, since natural language is ambiguous, but also due to the fact that the metathesaurus is built by merging different knowledge sources together, MetaMap

2. This utterance is from a small 50K abstract fraction of MEDLINE [14], having both *Hormones* and *Reproduction* as major topic keywords.

is not able to disambiguate all parts of the expressions. For instance, here *plasma, stimulation,* and *pregnancy* are all ambiguous. A simple solution to this problem is just to accept the ambiguity in the generation of descriptions, hence, produce all possible interpretations of the expressions, for instance, the two readings of *early pregnancy*

pregnancy{C0425965}[CHR:early{C1279919}]
pregnancy{C0032961}[CHR:early{C1279919}]

More advanced solutions could introduce additional methods for disambiguation of descriptions, for instance, try to include context analysis in order to further reduce the ambiguity, see [20] for a survey of word-sense disambiguation approaches. An example of the mapping of the concept pregnancy{C0032961}[CHR:early{C1279919}] into the UMLS is given in Figure 8.1.

Regardless of whether rules to combine into compound concepts are applied or not, the result of a mapping from a piece of text T to an ontology O will be a set of concepts. This set of concepts $d_O(T)$ we call the *description* of T (with respect to O) and the elements of d are called descriptors. $d_O(T)$ is, so to say, T viewed through the ontology O. The set of concepts $d_O(T)$ may be used as the content of an ontology-based indexing, for instance on the level of sentences. Here, our main focus is on summarization, and thus, we will also be concerned with descriptions covering larger texts and collections of texts. So all in all, no matter the size, form, or structure of a given text T, the basic description is a set of descriptors.

8.3.1 Instantiated Ontology

The description $d_O(T)$ of a text T, given the ontology O, comprises a set of concepts in O, and as indicated, the purpose here is to summarize based on relations in the ontology. Now given the set of concepts (the description) $d_O(T)$, an obviously relevant subontology is a subontology that covers all elements of $d_O(T)$. Such a subontology can be considered an instantiation of the text T (or the set of concepts $d_O(T)$). A very simple example on such ontology is the ontology for the concept pregnancy[CHR:early] given in Figure 8.1.

Given an ontology $O = (L, \leq, R)$ and a set of concepts C, we define the instantiated ontology $O_C = (L_C, \leq_C, R)$ as a restriction of O to cover only the concepts in C, that is, C and every concept from L that subsumes concepts in C or attributes for concepts in C. L_C can be considered an "upper expansion" of C in O. More specifically, with $C+$ being C extended with every concept related by attribution from a concept in C:

$$\begin{aligned} L_C &= C \cup \left\{ x \middle| y \in C^+, x \in L, y \leq x \right\} \\ \leq_C &= \left\{ (x, y) \middle| x, y \in L_C, x \leq y \right\} \end{aligned} \tag{8.5}$$

Thus O_C is not generative. $\leq_C$ may be represented by a minimal set $\leq$ $\leq'_C \subseteq \leq_C$ such that $\leq_C$ is derivable from $\leq'_C$ by means of transitivity of $\leq$ and monotonicity of attribution:

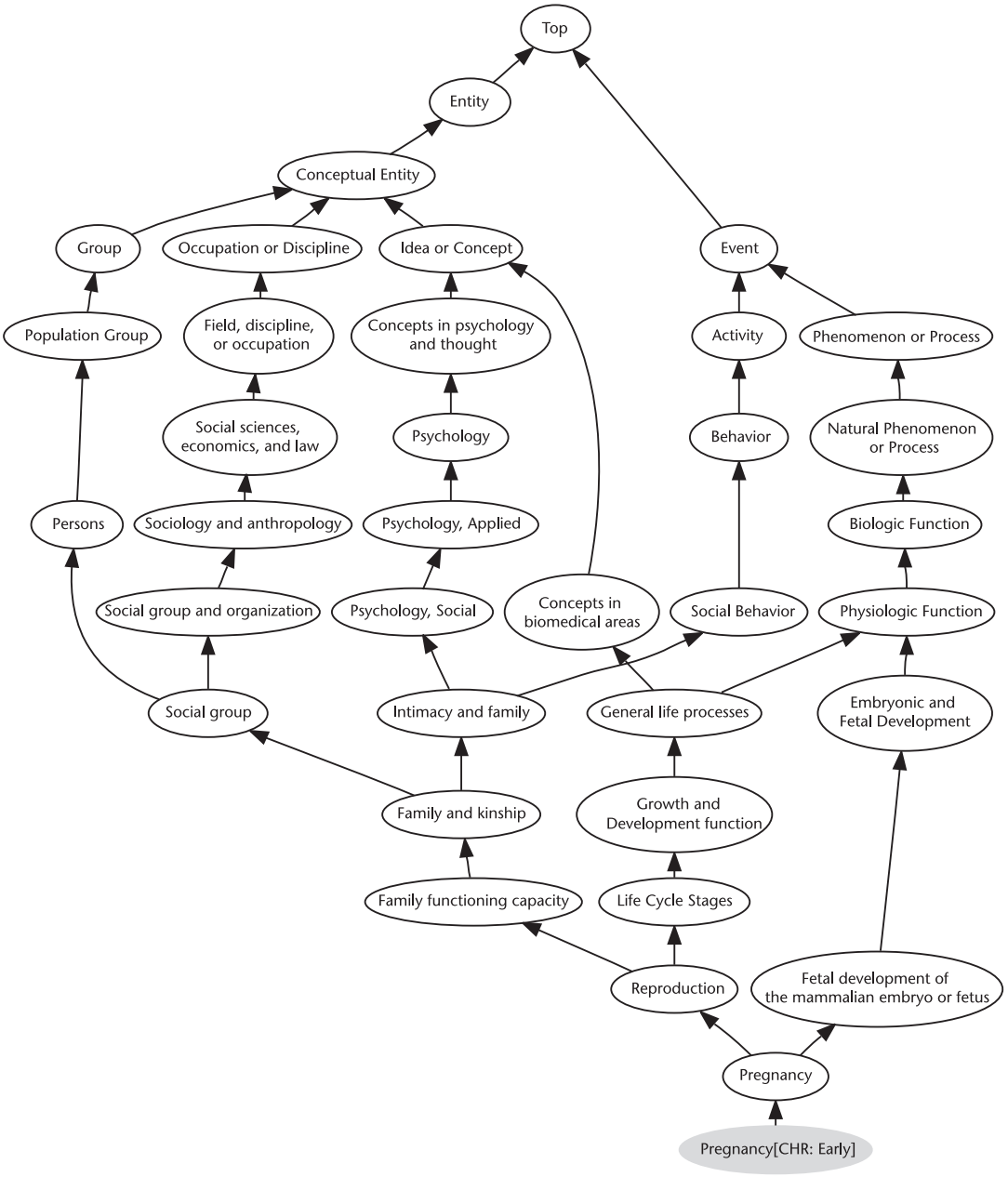


Figure 8.1 Mapping of the concept pregnancy{C0032961}[CHR:early{C1279919}] into UMLS (slightly modified, i.e., some paths are removed, due to considerations of space).

$$transitivity: x \le y, y \le z \Rightarrow x \le z$$
$$monotonicity: x \le y \Rightarrow z[r:x] \le z[r:y]$$

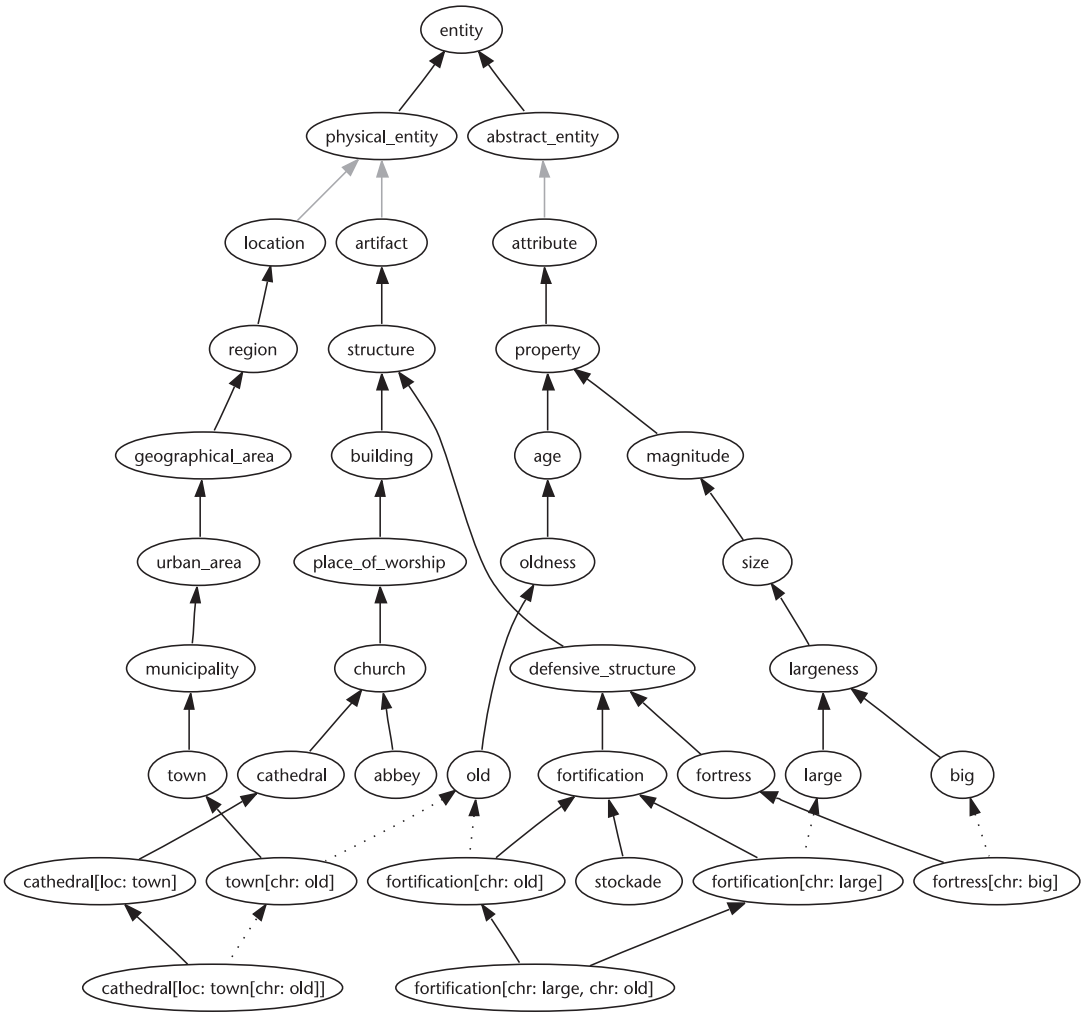


Figure 8.2 A segment of an ontology based on Wordnet that does also correspond to an instantiated ontology for the set of instantiated concepts {cathedral[LOC: town[CHR: old]], abbey, fortification[CHR: large, CHR: old], stockade, fortress[CHR: big]}.

Figure 8.2 shows an example of an instantiated ontology. The general ontology is based on (and includes) WordNet and the ontology shown is "instantiated" wrt. the following set of concepts:

$$C = \{cathedral[LOC : town[CHR : old]], abby,$$
$$fortification[CHR : large, CHR : old], stockade,$$
$$fortress[CHR : big]\}$$

8.4 Data Summarization Through Background Knowledge

The general idea here is to exploit background knowledge through conceptual summaries that are to provide a means to survey textual data, for instance, a query result. A set of concepts from the background knowledge is first identified in the text and then contracted into a smaller set of, in principle, the most representable concepts.

This can be seen as one direction in a more general conceptual querying approach, in which queries can be posed or answers be presented by means of concepts. For a general discussion on other means, except from conceptual summaries, of conceptual querying, where a dedicated language construct is presented for this purpose we refer to [8]. Here we discuss summaries only.

In the approach to summarization described here, we assume the use of an ontology to guide the summarization and, for the text to be summarized, an initial extraction of concepts, as described in Section 8.31. Thus, we can assume an initial set of concepts C and we a facing a challenge to provide a smaller set of representative concepts covering *C*, that is, an appropriate summary, that grasps what's most characteristic about *C*. For computation of the summary, we restrict to the subontology $O_C = (L_C , \leq_C , R)$, corresponding to the instantiated ontology for *C*.

One reasonable approach to providing summaries, along this line, is thus to divide the set of concepts into groups or clusters and to derive for each a representative concept—for instance the least upper bound (*lub*) for the group.

Throughout this section we will use one common example ontology, depicted in Figure 8.3, derived from one paragraph of text found in SEMCOR[10]:

> *Greases, stains,* and miscellaneous *soils* are usually sorbed onto the *soiled surface.* In most *cases,* these *soils* are taken up as *liquids* through *capillary action.* In an essentially static *system,* an *oil* cannot be replaced by *water* on a *surface* unless the *interfacial tensions* of the *water phase* are reduced by a *surface-active agent.*

where words in italics indicate the initial set of concepts, in this case nouns that are mapped into WordNet[11], from which the instantiated ontology is created[3]. SEMCOR is a subset of the documents in the Brown corpus [21], which has the advantage of being semantically tagged with senses from WordNet.

We introduce two directions for deriving summaries below: one based directly on connectivity in the ontology and the other drawing on statistical clustering applying similarity measures.

8.4.1 Connectivity Clustering

Connectivity Clustering is clustering based solely on connectivity in an instantiated ontology. More specifically the idea is to cluster a given set of concepts based on their connections to common ancestors, for instance grouping two siblings due to their common parent, and in addition to replace the group by the common ancestor. Thus rather than, when taking a bottom-up hierarchical clustering view, moving

3. Notice that due to the use of SEMCOR, there is no attribution in the initial set of concepts.

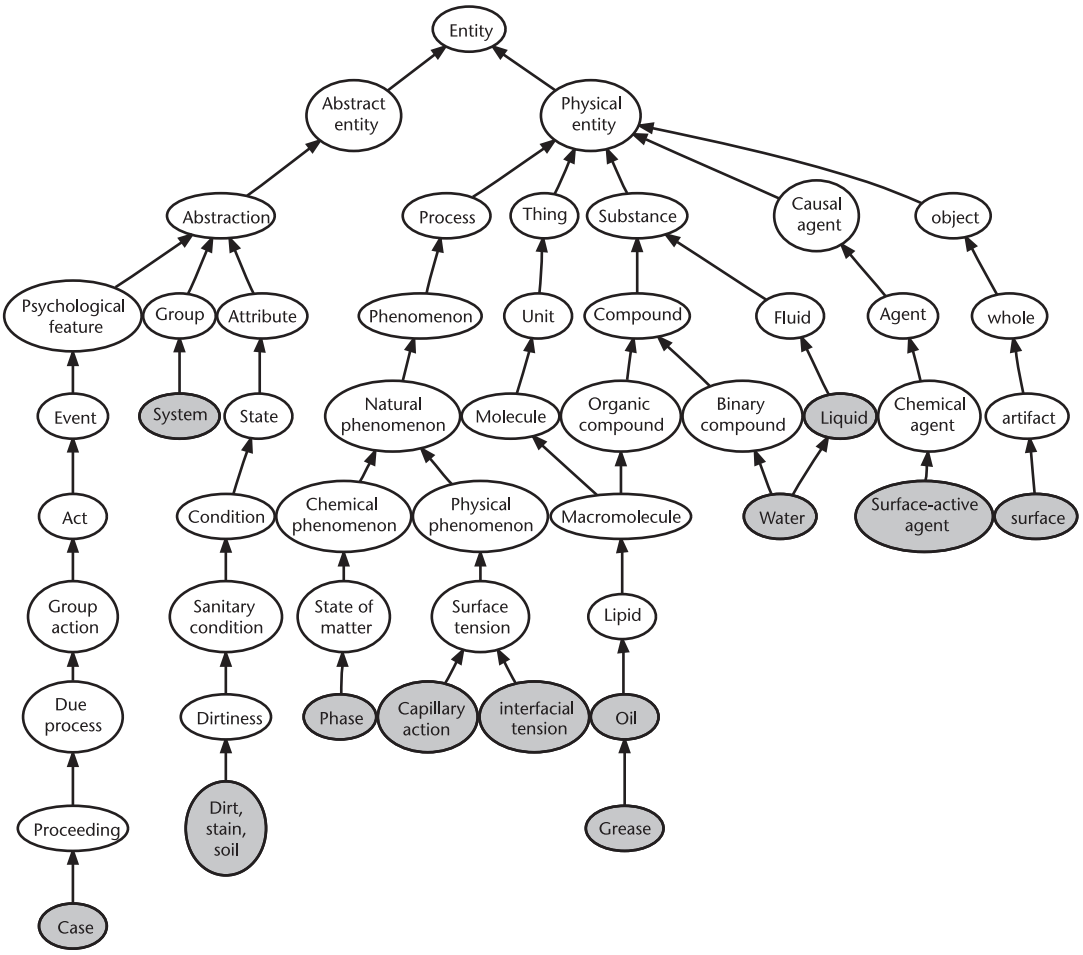


Figure 8.3 An instantiated ontology based on a paragraph from SemCor.

towards a smaller number of larger clusters, connectivity clustering is about moving towards a smaller number of more general concepts.

For a set of concepts $C = \{c_1, ..., c_n\}$ we can consider as generalizing description a new set of concepts $\delta(C) = \{\hat{c}_1, \cdots, \hat{c}_k\}$, where $\hat{c}_i$ is either a concept generalizing concepts in C or an element from C. Each generalizer in $\delta(C)$ is a least upper bound (lub) of a subset of C, $\hat{c}_i = \text{lub}(C_i)$, where $\{C_1, ..., C_k\}$ is a division (clustering) of C. Notice that the lub of a singleton set is the single element in this.

We define the most specific generalizing description $\delta(C)$ for a given $(c) = \{\hat{c}_1, \cdots, \hat{c}_k\}$ as a description restricted by the following properties:

$$\forall \hat{c} \in \delta(\mathrm{C}) : \hat{c} \in \mathrm{C} \vee \exists c', c'' \in \mathrm{C} \wedge c' \neq c'' \wedge c' < \hat{c} \wedge c'' < \hat{c} \tag{8.6}$$

$$\forall \hat{c}', \hat{c}'' \in \delta(\mathrm{C}) : \hat{c}' \not< \hat{c}' \tag{8.7}$$

$$\forall c', c'' \in C, \hat{c}' \in \delta(C), \neg\exists x \in L_C : c' \leq x \wedge c'' \leq x \wedge x \leq \hat{c}' \tag{8.8}$$

where (8.6) restricts $\delta(C)$ to elements that either originate from C or generalize two or more concepts from C. Equation (8.7) restricts $\delta(C)$ to be without redundance (no element of $\delta(C)$ may be subsumed by another element), and (8.8) reduces to the most specific, in the sense that no subsumer for two elements of C may be subsumed by an element of $\delta(C)$.

Observe that $\delta(C)$, like C, is a subset of L_C, and that we therefore can refer to an m'th order summarizer $\delta^m(C)$. Obviously, to obtain an appropriate description of C, we will in most cases need to consider higher orders of δ. At some point m, we will in most cases, have that $\delta^m(C) = Top$, where Top is the topmost element in the ontology. An exception is when a more specific single summarizer is reached in the ontology.

The most specific generalizing description $\delta(C)$ for a given C is obviously not unique, and there are several different sequences of most specific generalizing descriptions of C from C toward Top. However, a reasonable approach would be to go for the largest possible steps obeying the restrictions for δ above, as done in the algorithm below.

For a poset S, we define $min(S)$ as the subset of minimal elements of S: $min(S) = \{s | s \in S, \forall s' \in S : s' \not< s\}$

ALGORITHM—Connectivity summary.

INPUT: Set of concepts $C = \{c_1, ..., c_n\}$.

OUTPUT: A most specific generalizing description $\delta(C)$ for C.

1. Let the instantiated ontology for C be $O_C = (L_C, \leq_C, R)$
2. $U = min(\{u | u \in L_C \wedge \exists c_i, c_j \in C : c_i < u \wedge c_j < u\})$,
3. $L = \{c | c \in L_C \wedge \exists u \in U : c < u\}$
4. $M = min(\{m | m \in L_C \backslash L \wedge \exists c \in L : c < m\})$,
5. set $\delta(C) = C \cup U \cup M/L$

In 2, all of the most specific concepts U that generalize two or more concepts from C are derived. Notice that these may include concepts from C when C contains concepts subsuming other concepts. In 3, L defines the set of concepts in C that specializes the generalizers in U. In 4 additional parents for (multiple inheriting) concepts covered by generalizations in U are added. 5, derives $\delta(C)$ from C by adding the most specific generalizers and subtracting concepts specializing these.

Notice especially that 4, is needed in case of multiple inheritance. If a concept C has multiple parents and is replaced by a more general concept due to one of its senses (parents) we need to add parents corresponding to the other senses of C—otherwise, we lose information corresponding to these other senses. For instance, in Figure 8.4 we have that $\delta(\{a, b\}) = \{c, d\}$ because a and b will be replaced by c, and d will be added to specify the second sense of b.

As a more elaborate example consider again Figure 8.3. Summarization of C by connectivity will proceed as follows.

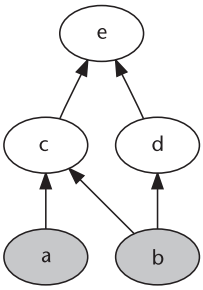


Figure 8.4 Ontology fragment with multiple inheritance.

C = *{case, system, dirt, phase, capillary action, interfacial tension, grease, oil, water, liquid, surface-active agent, surface}*

$\delta^1(C)$ = *{abstraction, binary compound, liquid, oil, phase, surface, surface-active agent, surface tension}*

$\delta^2(C)$ = *{abstraction, compound, liquid, molecule, natural phenomenon, surface, surface-active agent}*

$\delta^3(C)$ = *{abstraction, molecule, natural phenomenon, substance, surface, surface-active agent}*

$\delta^4(C)$ = *{abstraction, physical entity}*

$\delta^5(C)$ = *{entity}*

The chosen approach, taking the largest possible steps in which everything that can, will be grouped, is, of course, not the only possible approach. If we, alternatively, want to form only some of the possible clusters complying with the restrictions, some kind of priority mechanism for selection is needed.

Among important properties that might contribute to priority are *deepness, redundancy,* and *support*. The deepest concepts, those with the largest depth in the ontology, are structurally, and thereby often also conceptually, the most specific concepts. Thus, collecting these first would probably lead to a better balance with regard to how specific the participating concepts are in candidate summaries. Redundancy, where participating concepts include (subsume) others, is avoided as regards more general concepts introduced (step 3 in the algorithm). However redundancy in the input set may still survive so priority could also be given to remove this first. In addition we could consider support for candidate summarizers. One option is simply to measure support in terms of the number of subsumed concepts in the input set while more refinement could be obtained by also taking the frequencies of concepts, as well as their distribution in documents[4] in the original text into consideration. Support may guide the clustering in several ways. It indicates for a concept how much it covers in the input and can thus be considered as an importance weight for the concept as summarizer for the input. High importance should probably infer more reluctance, as regards further generalization.

4. Corresponding to term and document frequencies in information retrieval.

8.4.2 Similarity Clustering

While we may consider connectivity clustering as ontology-based in a genuine sense it is not the only possible direction. Alternatively, cluster applying, given similarity measures, over the set of concepts should also be considered. Obviously, if the measure is derived from an ontology, and thereby does reflect this, then so will the clustering. Below we will assume an ontology-based similarity measure *sim*, as briefly touched upon in Section 8.2.3 but, make no further assumptions of the type and characteristics of this measure.

We may expect a pattern similar to connectivity clustering in the derivation of summaries in an approach based on similarity, when the similarity measure closely reflects connectivity in the ontology, as the simple shortest path measure does.

Beside the example ontology from Figure 8.3 we will use the following set of clusters:

$\{C_1$ = *{size, number}*
C_2 = *{government, state, committee}*
C_3 = *{defender, man, servant, woman}*
C_4 = *{cost, bribe, price, fee}*
C_5 = *{fortress, fortification, stockade}}*

where C_1,...,C_4 are from SemCor and C_5 is from the example ontology in Figure 8.2. The former is created by use of a categorical agglomerative hierarchial clustering, with the shortest path similarity measure [12] over all documents in SemCor containing the word *jury*. We then picked from one of the levels in the clustering some meaningful clusters, and furthermore added a structure cluster (*fortress, fortification, stockade*) from Figure 8.2 to capture another aspect.

8.4.2.1 A Hierarchical Similarity-Based Approach

With a given path-length dependent similarity measure derived from the ontology, a *lub*-centered, agglomerative, hierarchical clustering can be performed as follows.

Initially each cluster corresponds to an individual element of the set to be summarized. At each particular stage, the two clusters that are most similar are joined together. This is the principle of conventional hierarchical clustering; however rather than replacing the two joined clusters with their union, as in the conventional approach they are replaced by their *lub*. Thus, given a set of concepts $C = \{c_1, ..., c_n\}$, summarizers can be derived as follows.

ALGORITHM—Hierarchical clustering summary.
INPUT: Set of concepts $C = \{c_1, ..., c_n\}$.
OUTPUT: Generalizing description $\delta(C)$ for C.

1. Let the instantiated ontology for C be $O_C = (L_C , \leq_C , R)$
2. Let $T = \{(x, y) | sim(x, y) = \max_{z,w \in C} (sim(z, w))\}$,
3. Let $U = min(\{u | u \in L_C \wedge \exists x, y \in L_C : x<u \wedge y<u\})$,
4. $L = \{x | <x, y> \in T \vee <y, x> \in T\}$
5. set $\delta(C) = C \cup U/L$

As was also the case with the connectivity clustering, to obtain an appropriate description of C we might have to apply δ several times and at some point m we have that $\delta^m(C) = Top$.

Figure 8.5 illustrates the application of δ for a total of 15 times to the set of concepts from the previous example. At each step is shown the lub with which the two clusters is replaced. Thus, the first step is to replace *cost* and *price* with *cost,* the second to replace *government* and *state* with *government,* and so on.

8.4.2.2 Simple Least Upper Bound-Based Approach

A straightforward similarity based approach is simply to apply a crisp clustering to the set of concepts $C = \{c_1, ..., c_n\}$ leading to $\{C_1,...,C_k\}$ and then provide the set of $\{\hat{c}_1, \cdots, \hat{c}_k\} = \{lub(C_1), ..., lub(C_k)\}$ for the division of C as summary[5] to also take into account the importance of clusters in terms of their sizes the summary can be modified by the support of the generalizing concepts, $support(x, C)$, that for a given concept specifies the fraction of elements from the set C covered:

$$support(x, C) = \frac{\left|\{y | y \in C, y \leq x\}\right|}{|C|} \tag{8.9}$$

leading to a fuzzyfied (weighted) summary, based on the division (crisp clustering) of C into $\{C_1,...,C_k\}$:

$$\sum_i support(lub(C_i), C) / lub(C_i) \tag{8.10}$$

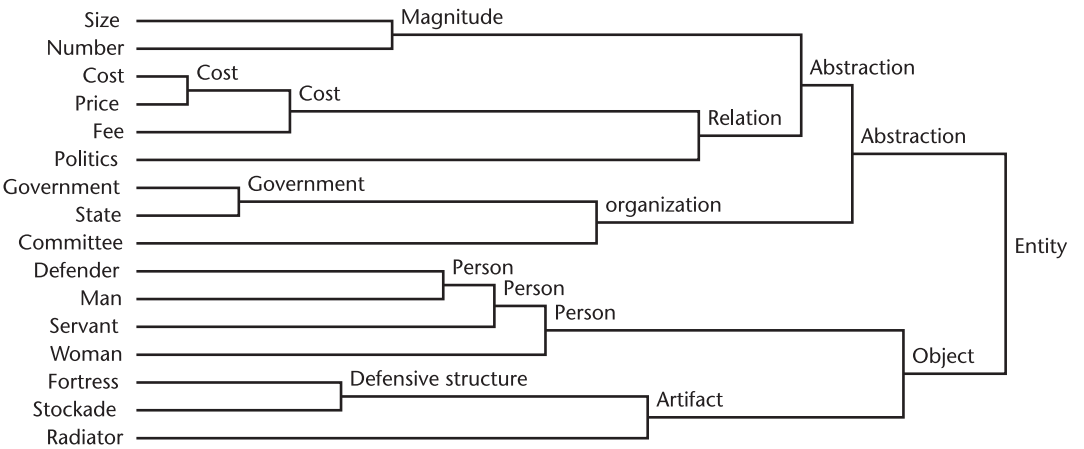


Figure 8.5 An illustration of the hierarchical clustering summary. The merging of two clusters is shown with their lub.

5. Observe that we put no restrictions on the clustering here, but, of course, the general idea is that the clustering applies a similarity measure that is ontology-based and that the ontology reflected is the instantiated ontology over the set of concepts C.

Table 8.1 A Set of Crisp Clusters and Their Least Upper Bounds from WordNet

Cluster	*Lub*
{number, size}	Magnitude
{committee, government, state}	Organization
{defender, man, servant, woman}	Person
{bribe, cost, fee, price}	Cost
{fortification, fortress, stockade}	Defensive structure

To illustrate this lub-based approach, consider the set of clusters and their least upper bounds in Table 8.1.

From these clusters the fuzzyfied summary {.13/*magnitude* + .19/*organization* + .25/*person* + .25/*cost* + .19/*defensive structure*} can be generated. Obviously, this approach to summarization will not be very tolerant as regards noise in the clusters given. Consider the following example, in which *bribe* is replaced by *politics* and *stockade* by *radiator*. The least upper bounds of the respective clusters becomes more general as illustrated in Table 8.2.

As a result, the summary becomes {.13/*magnitude* + .19/*organization* + .25/*person* + .25/*relation* + .19/*artifact*}. To get around this problem, we can introduce a soft definition of lub and combine this again with crisp clusters to get more specific cluster-based summaries. Obviously, this approach to summarization will not be very tolerant as regards noise in the clusters given. The *lub* for a cluster including an outlier may well be the top element or something similarly useless.

8.4.2.3 A Soft Least Upper Bound Approach

To get around this problem we can introduce a soft definition of *lub* and combine this again with crisp clusters to get more specific cluster-based summaries.

A soft definition of *lub* for a (sub)set of concepts C' should comprise *upper boundness* as well as *leastness* (or *least upperness*), expressing, respectively the portion of concepts in C' that are generalized and the degree to which a concept is least upper with regard to one or more of the concepts in C'.

Upper boundness can be expressed for a set of concepts C' by $\mu_{ub(C')}$ simply as the support as regards C':

Table 8.2 A Set of Crisp Clusters with Noise and Their Least Upper Bounds from WordNet

Cluster	*Lub*
{number, size}	Magnitude
{committee, government, state}	Organization
{defender, man, servant, woman}	Person
{cost, fee, **politics**, price}	Relation
{fortification, **radiator**, stockade}	Artifact

$$\mu_{ub(C')}(x) = support(x, C') \tag{8.11}$$

covering all generalizations of one or more concepts in C' and including all concepts that generalize all of C' (including the topmost concept *Top*) as full members.

Leastness can be defined on top of a function that expresses how close a concept is to a set of concepts C' such as $dist(C', y) = min_{x \in C'} dist(x, y)$, where $dist(x, y)$ expresses the shortest path upwards[6] from x to y, as follows:

$$\mu_{lu(C,\lambda)}(x) = \begin{cases} 1 & \text{when } \lambda = 0 \vee x = Top \\ 1 - \dfrac{dist(C', x)}{dist(C', Top) + \dfrac{1}{\lambda} - 1} & \text{otherwise} \end{cases} \tag{8.12}$$

where $0 \leq \lambda \leq 1$ is a leastness parameter, with $\lambda = 1$ corresponding to the most restrictive version of leastness and with the other extreme $\lambda = 0$ corresponding to no restriction at all (all upper concepts become full members).

A simple soft least upper bound flub can now be defined as the product between μ_{lu} and μ_{ub}

$$\mu_{flub(C',\lambda)}(x) = \mu_{lu(C',\lambda)}(x) * \mu_{ub(C')}(x) \tag{8.13}$$

Notice that a lub for C is not necessary a best candidate among the flub elements. Thus, again with a division (crisp clustering) of C into $\{C_1, \ldots, C_k\}$, the basis for the summary here is the set of fuzzy sets $\{flub(C_1), \ldots, flub(C_k)\}$leading to the summary

$$\delta(\{C_1, \cdots, C_k\}) = \left(\bigcup_{i=1}^{k} flub(C_i) \right) \tag{8.14}$$

As in the lub-based case the summarizers should, in addition, be weighted by support. Thus, we can weight all elements in each $\{flub(C_1), \ldots, flub(C_k)\}$ with the support of $flub(C_i)$ in C:

$$\delta(\{C_1, \cdots, C_k\}) = \left(\bigcup_{i=1}^{k} flub(C_i) \right) \otimes \left(\sum_{x \in flub(C_i)} \frac{|(C_i)|}{|C|} \Big/ x \right) \tag{8.15}$$

where $\otimes$ is a *t*-norm, probably with product as an appropriate choice. Finally this set should obviously be restricted by some appropriate threshold α:

$$\delta_\alpha(\{C_1, \cdots, C_k\}) = \{m/x | m/x \in \delta(\{C_1, \cdots, C_k\}) \wedge m > \alpha\} \tag{8.16}$$

6. Upward refers to the idea that only paths consisting soely of edges in the direction of should be taken into account, and it should be strictly emphasized that the graph considered correspoinds to the transitively reduced ontology.

Given the previous example of noisy crisp clustering the use of a flub-based summary[7] will lead to

{.19/*cost* + .15/*outgo* + .15/*relation*+
.15/*person* + .13/*organization* + .13/*government*+
.11/*finacialloss* + .11/*artifact*+
.11/*defensivestruture* + ...}

while the lub based summary was

{.13/*magnitude* + .19/*organization*+
.25/*person* + .25/*relation* + .19/*artifact*}

In the *flub* based summary, *cost* has a high degree of membership, due to the fact that it is a very good description of three of the four elements in the cluster {*cost*, *price*, *fee*, *politics*}. Thus, the introduction of the *flub* reduces the effect of noise caused by the noisy element politics. Also, the degree of membership of artifact is comparable to the degree of membership of *defensive structure*, which is the immediate generalization of *fortress* and *stockade*. Again, the result of using a flub based summary is that the effect of the noisy element radiator in the cluster {*fortress*, *stockade*, *radiator*} is reduced.

8.5 Conclusion

In this chapter we have considered how to use ontologies to provide data summaries, with a special focus on textual data. Such summaries can be used in a querying approach where concepts describing documents, rather than documents directly, are retrieved as query answers. The summaries presented are conceptual, due to fact that they exploit concepts from the text to be summarized, and ontology-based because these concepts are drawn from a reference ontology.

The principles for conceptual summarization are presented here, as related to so-called instantiated ontologies—a conceptual structure reflecting the content of a given document collection, and therefore, particularly well suited as a target for conceptual querying; however, the summaries introduced are not dependent on this notion.

We have discussed some possible directions and basic principles for summarizations. It is obvious that development of more specific turnkey methods for summarization should be guided by profound experiments within a framework that includes a realistic general world knowledge resource. Initial studies have been done with a WordNet-based general ontology and SEMCOR as corpus (Information base); and preliminary experiments have been done with an UMLS-based general ontology and with a selection of abstracts from MedLine [14] as corpus; however, much more thorough experimental work needs to be done.

In addition, summary evaluation principles should be taken into account, first of all, as complement in guiding the development of summary principles. However, specific characteristics encircling good summaries may also be used in solutions to

7. With λ being 1.

the stopping problem—on deciding when, in the process of incremental contraction of (potential summary) sets, the best candidate has been found.

Initial considerations on the quality of summaries can be found in [19], but the issue is also an obvious direction for further work in continuation of what has been described here.

References

[1] Jensen, P. A. and J. F. Nilsson, "Ontology-Based Semantics for Prepositions, in Syntax and Semantics of Prepositions," P. Paint-Dizier (ed.), *Text, Speech and Language Technology,* Vol. 29, Springer, 2006.

[2] Aronson, S.R., "Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap Program," *Proc AMIA Symp,* 2001, pp. 17–21.

[3] Abney, S., "Partial Parsing via Finite-State Cascades," *Proc of the ESSLLI '96 Robust Parsing Workshop,* 1996.

[4] Bodenreider, O., "The Unified Medical Language System (UMLS): Integrating Biomedical Terminology," *Nucleic Acids Research,* Vol. 32, 2004, pp. D267-D270.

[5] Hahn, U., and I. Mani, "The Challenges of Automatic Summarization," *Computer,* November 2000.

[6] Melli, G., et. al., "Description of SQUASH, the SFU Question Answering Summary Handler for the DUC-2005 Summarization Task," *Proc of DUC-2005,* Vancouver, Canada, 2005, pp. 103–110.

[7] Shi, Z., et al., "Question Answering Summarization of Multiple Biomedical Documents," *Lecture Notes in Computer Science,* Springer Berlin/Heidelberg Vol. 4509/2007, *Advances in Artificial Intelligence,* 2007, pp. 284-295.

[8] Andreasen, T., H. Bulskov, "Conceptual Querying Through Ontologies," *Fuzzy Sets and Systems,* Vol. 160, No. 51.

[9] Nilsson, J.F., "A Logico-Algebraic Framework for Ontologies—ONTOLOG," in Jensen, P.A., Skadhauge, P., (eds.), *First Int. OntoQuery Workshop,* University of Southern Denmark, 2001.

[10] Miller, G.A., et. al., "Using a Semantic Concordance for Sense Identification," *Proc. of the ARPA Human Language Technology Workshop,* 1994, pp. 240–243.

[11] Miller, G.A., "Wordnet: A Lexical Database for English," *Commun. ACM,* Vol. 38, No. 11, 1995, pp. 39–41.

[12] Rada, R., et. al., "Development and Application of a Metric on Semantic Nets," *IEEE Trans. on Systems, Man, and Cybernetics,* Vol. 19, No. 1, January 1989, pp. 17–30.

[13] Unified Medical Language System, U.S. National Library of Medicine, http://www.nlm.nih.gov/research/umls/

[14] Medical Literature Analysis and Retrieval System Online, U.S. National Library of Medicine, http://www.ncbi.nlm.nih.gov/pubmed/.

[15] Bulskov, H., R. Knappe, and T. Andreasen, "On Measuring Similarity for Conceptual Querying," *FQAS '02: Proc. of the 5th Inter. Conf. on Flexible Query Answering Systems,* Springer-Verlag, 2002, pp. 100–111.

[16] Resnik, P., "Semantic Similarity in a Taxonomy: An Information-Based Measure and Its Application to Problems of Ambiguity in Natural Language," 1999.

[17] Andreasen, T., R. Knappe, H. Bulskov, "Domain-Specific Similarity and Retrieval," *Proc. IFSA 2005,* Tsinghua University Press, 2005, pp. 496–502.

[18] Andreasen, T., et. al., "Content-Based Text Querying with Ontological Descriptors," *Data Knowledge Engineering,* Elsevier Science Publishers, Vol. 48, No. 2, pp. 199–219.

[19] Yager, R. R., F. E. Petry, "A Multicriteria Approach to Data Summarization Using Concept Hierarchies," *IEEE Trans. on Fuzzy Sys.,* Vol. 14, No. 6, 2006.

[20] Zhou, X., H. Han, "Survey of Word Sense Disambiguation Approaches," *18th FLAIRS Conference,* 2005.

[21] Francis, W. N. and H. Kucera, *Brown Corpus Manual,* Department of Linguistics, Brown University, 1964.

CHAPTER 9

Reasoning over Anatomical Ontologies

Toni Kazic, Jennifer L. Leopold, and Anne M. Maglia

9.1 Why Reasoning Matters

Discovering new biological knowledge requires data mining—accessing, integrating, and analyzing potentially heterogeneous data. All these steps require the expertise and reasoning of trained minds. In principle, all are a continuum of activities, ranging from the fully manual to, one hopes someday, the fully automatic. As the automation level of data-discovery processes increases, a corresponding sophistication of the reasoning that can be conducted may also increase. But today, sophistication suddenly implodes at fully automated operations. Moreover, the most sophisticated automated operations revolve around sequence and structure, which have received the greatest attention of computational biologists in the last 40 years.

For many of the most important biological questions, however, such as species identification, ecosystem processes, and epigenesis, molecular sequence and structure are only part of the information biologists need for their analyses. While our ability to collect, archive, and disseminate various kinds of data has exploded, people remain the bottleneck in knowledge discovery: data resources far outnumber the available experts, even as the complexity and urgency of the questions we must answer increase.

Accelerating the rate at which we can answer biological questions is no longer an abstract academic concern. Consider the crises facing taxonomists, geneticists, and developmental biologists. Only about 1.8 million of the estimated 10 million species on Earth have been described; nearly 20% of the known species in some taxa are headed for extinction, and many unknown species may be extinct before they can be described [1]. Unfortunately, the time required to describe species using traditional methods precludes the rapid cataloging of biodiversity. The rate at which we understand organismal development, especially for the crop plants that sustain humans and their domestic animals, may well determine our collective future. Environmental degradation, exploding populations, mushrooming energy demands, and the inexorable sprawl of pavement over croplands all intensify the need to better understand and manipulate phenotypes, such as yield, environmental impact, drought tolerance, disease resistance, and efficiency of production. Genetics and the thoughtful exploitation of genomic and postgenomic information are central to the identification of contributing genes and understanding their mechanisms and

interactions in producing these complex phenotypes. The best algorithms remain only suggestive, however, and biologists still sift data manually, comparing it to the literature and thinking through hypotheses and their possible tests.

How can we accelerate discovery? The obvious answer is to represent expert knowledge in ways that permit algorithms to emulate expert reasoning. Biodiversity, genetics, and developmental biology require the comparison and analysis of phenotypic information from *many individuals of different species*. Genetics and developmental biology also rest on wild-type and mutant phenotypes within a species. Variations within species illuminate the mechanisms of variations among species when ideas from morphology, genetics, ecology, function, genomics, and phylogeny are combined. All three fields entail very complex, interdisciplinary reasoning over huge bodies of richly detailed data. A reasoning system that could automatically identify and describe new species by recognizing, encoding, and comparing precisely defined phenotypic characteristics of *many individuals* would accelerate and increase the reliability of species identification and generate data for evolutionary, phylogenetic, and phylogeographic analyses. Applying the same techniques to mutants from a single species would increase the precision of phenotypic descriptions, help organize genes into processes and mechanisms, and help pinpoint the effects of developmental mutants. Properly defined and rigorously tested on masses of real data, automating biological reasoning would open new frontiers.

In this chapter, we explore some of the representational issues that automating biological reasoning raises. The wide dissemination of ontologies makes them an excellent starting point for discussion for several reasons. First, ontologies have become one of the most widely distributed representations of biological knowledge. Second, the choices made by ontology builders highlight some of the pitfalls in representing biology for automated reasoning. Third, they are ideal laboratories for understanding the needs of reasoning systems aimed at primary biological data. After all, in computer science, ontologies were originally conceived as structured knowledge represetations for use in reasoning. Finally, ontologies have already been used in reasoners, and more examples are sure to come. We focus on anatomical information from maize and on the ideas about phenotypes, function, development, and phylogeny that arise when experts contemplate specimens. This is not to imply that all of the issues of reasoning with molecular sequences and structures have been solved—far from it! But to create more sophisticated knowledge-discovery tools, we must apply them to more diverse, complex biological systems.

We begin by describing the characteristics of different types of data and the reasoners that have used it, to date. We then describe anatomical data and explain why it is a challenging platform for the development of reasoners. In Section 9.3, we describe some of the current practices in building ontologies, the impact they can have on representing anatomy, and the issues raised for anatomical reasoning. We use the maize tassel as an example, presenting a brief outline of its anatomy, phenotypes, and development. In Section 9.4, we describe several possible approaches to ameliorating the issues, and conclude with some possible future directions. Our subject necessarily involves ideas, literature, and trends from many different areas, and we cannot hope to be remotely complete. Thus, the reader will find several recent reviews, special journal issues, and the introduction of many research papers helpful in forming his or her own picture of this rapidly changing area [2–7].

9.2 Data, Reasoning, and a New Frontier

9.2.1 A Taxonomy of Data and Reasoning

We begin by considering the types of data available, as a function of their proximity to experimentation and observation. Not surprisingly, the type of data strongly influences the type of reasoning one can undertake.

Logicians distinguish between *extensional* data—data that is explicitly stated and forms the axioms of a reasoning system—and *intensional* data—data that is inferred from extensional data, using logical reasoning [8]. Heuristically, extensional data is the totality of biological facts that one couldn't have imagined prior to their discovery, such as the rate of a reaction under particular conditions or the number of recombinants in a particular cross. In contrast, intensional data is derived from the extensional data—the enzyme's K_m or the map distance between the markers. In the taxonomy we next present, which is ordered by the proximity of the data to a direct observation, data that is intensional at one level often becomes extensional at the next, for reasons of representational or computational convenience.

9.2.1.1 Primary Data

This data arises from direct observation, either in laboratory experiments or in the field, clinic, or society. Thus, it is extensional. Examples include raw image data files collected from CT scans or photographs of organisms; the trace of an automated DNA-sequencing machine or gas chromatograph; the count of wild-type and mutant offspring in a genetic cross; or an autoradiograph of a blot. Heuristically, this is the data reported in the results sections of papers and the data that one interprets. There are many, many programs that reason with primary data, and a short list can be only illustrative. Examples include MAPMAKER, which constructs genetic maps from cross data; PHRED, which assigns bases in the DNA sequence from the raw chromatographic trace; DIRDIF and SOLVE, which solve crystal structures from the diffraction data; and software that white-balances, color-corrects, and renders images from the digital camera's detector output, such as the software packaged in the camera or Gimp and Me*VisLab* [9–16].

9.2.1.2 Derived Data

This is the interpretation of primary data—a genetic map, a species, an hypothesis about the mechanism of a reaction, the mode of action of a gene, or the outputs of qualitative or quantitative models. This is the intensional data usually presented in the discussion or conclusions sections of papers. Examples for polymers include programs that assemble longer sequences from those of fragments or predict the solvent accessibility of proteins [17, 18]. Nonpolymer examples include the work of Shortliffe diagnosing bacterial infections from laboratory values of patients [19]; Karp emulating Yanofsky's studies of the *trp* operon of *Escherichia coli* [20]; Altman et al. drawing inferences about ribosomal structure and function [21]; and Pearl studying the reasoning behind epidemiological studies [22]. Another nonpolymeric example is Lucid3, an expert system that generates taxonomic keys from

user-supplied information [23]. All this work draws inferences from primary or derived data, or a mix of the two, by emulating biologists' reasoning. (By permitting a bit of elasticity, one can lump primary and derived data together under the term of *observational data*.) It is noteworthy that the interpretations are often unambiguous, become relatively uncontroversial rather rapidly, and are robust to changes in conditions or parameters.

Recently, several systems have been built that serve ad hoc biological questions, by collecting and analyzing molecular data and results from disparate resources over the Web [24–31]. All leverage Semantic Web or Grid technologies [6, 32], incorporate some reasoning in query formulation and execution, and are based on the idea that a user executes a connected network of tasks and analyses, or `workflow`. Several of these also support certain kinds of domain-based reasoning over the observational data [24, 25, 29]. The recently inaugurated iPlant project envisions the creation of "discovery environments," each a platform for investigating a set of biological questions, though the data will not be just molecular [33]. Discovery environments appear to have many ideas in common with workflows. All of these efforts provide early examples of what possible future reasoners may become for nonmolecular data.

9.2.1.3 Amalgamated Data

This is intensional data, which is produced by summarizing, integrating, and comparing data from multiple sources to produce a more systematic or comprehensive interpretation of natural phenomena. Examples include review articles, databases, metabolic and other pathway charts, phylogenetic trees, classifications of taxa and enzymes, maps of syntenous regions among homologous chromosomes from different organisms, comparisons of mathematical models simulated with different sets of parameter values, and ecological food webs [34–46]. Programs that reason with such data include any of the sequence or structure similarity and alignment programs and the related ones that compile substitution matrices or short motives correlated with particular functions; programs that construct phylogenetic trees; and programs that detect and align syntenous regions of chromosomes [9, 47–61]. Conclusions derived from algorithms can be altered by changing the values of the parameters used in the computation; those from human effort can be controversial among experts for significant lengths of time.

9.2.1.4 Annotation Data

The last category is the data used to annotate or categorize other types of data in compendia, especially databases. The primary examples here are controlled vocabularies, nomenclature, library cataloging systems, and ontologies [39–41, 62–70]. All strive to constrain the inventiveness and elasticity of language into a standardized group of unambiguously defined words. The use of ontologies to represent biological ideas and annotate databases has snowballed in the last 10 years [62, 63, 67, 71]. Ontology terms are short phrases drawn from the customary biological vocabulary that are assumed to directly communicate their sense to a trained biologist. The Open Biological Ontologies (OBO) Foundry is attempting to standardize

the structure and content of the contributed ontologies and to serve as a model for best practices in ontology building [67]. Such models and practices are extremely valuable, as building expressive and efficient ontologies remains an art form [7, 72, 73]. In this chapter, we focus on the representational and technical practices common in building ontologies, rather than on the social practices that provide the context for building; the latter are described in [2].

For our purposes, we define biological ontologies as collections of controlled vocabulary, organized into graphs that denote biological ideas, that often reflect a consensus archetype of one or more species. What distinguishes an ontology from a controlled vocabulary or nomenclature is that it organizes its terminology (the noun phrases) with a set of operators denoting ideas about biological or logical relationships (the verb phrases) of the system, producing a set of structured assertions about the ontology's subject. In both terminology and topology, ontologies reflect the considered scientific opinion of their builders, as they interpret the other types of data, and their use by people or algorithms in annotating data in databases applies those interpretations. For reasoners, the interpretative step is key; without knowing why a particular interpretation is assigned to data, a resonator has no substantive way to independently judge the data's correctness or utility for the reasoning task.

The difference between reasoning over the data on which the interpretations are based and the interpretations is like the difference between an analysis of the topology of the lunar surface and an analysis of a drawing of a moon crater. That said, reasoners that operate with *both* ontologies and observational data could be faster than reasoners that operate only over the observational data. The considered abstractions of specialist curators could be used to restrict the search space or recover alternative hypotheses for consideration, along with those induced from the observational data.

9.2.2 Contemporary Reasoners

The growth in terminological standardization of anatomy clearly demonstrates the interest in using ontologies for data management and "manual" knowledge discovery. Not surprisingly, several groups are attempting to develop biological reasoners that use ontologies. So far, these have one of two goals: either to check the consistency of the terms or to infer additional instances of extensional relationships among them. Several groups have developed reasoners for ontologies, including Pellet, Protégé and its plug-ins, Racer, and the OBO-Edit reasoner [74–80]. When applied to biological ontologies—for example, the Unified Medical Language System (UMLS), the Foundational Model of Anatomy (FMA), and the Gene Ontology (GO)—these systems are currently limited to checking the integrity of the ontologies [63, 66, 71, 81]. Logical operations that accommodate the structure and constraints of OBO and its predecessor ontologies have been defined [82–84]. This work can be seen as part of broader efforts to define reasoners that operate over the languages proposed for the Semantic Web, such as Description Logic programs, RuleML, SWRL, Datalog, SHOIN, and many others [4, 85]. A core technology of the Semantic Web is ontologies, which are intended to provide semantic interoperability among disparate resources, so reasoning over such resources is a very

active research area [86]. Space prevents us from surveying this area, though we try to provide some clues below to sources of information. Somewhat orthogonal to work on general reasoning languages are efforts to define representative predicates that permit spatial reasoning for anatomy [87]. The latter has led to a clearer understanding of other relations that would enrich and clarify existing anatomical ontologies if they were included.

9.2.2.1 Protégé

Protégé is an ontology-construction system [88–90]. It supports ontology maintenance by checking set membership (e.g., pepperoni is a part of a pizza and a member of the class of sausages); testing for necessity and sufficiency (e.g., a crust is necessary for a pizza); and by detecting instances that violate constraints (e.g., that a roulade is not a pizza because it doesn't have a crust) [92]. It can't explicitly say that "some pizzas have pepperoni." While true and valuable in some cases, this is a statement of modal logic. This shortcoming can be particularly problematic for biology, since most individuals do not exactly match an archetype [7]. There are currently six plug-in reasoners that can operate on Protégé-built ontologies, in either the frame or OWL representation, including *Jess* and SWRL [85, 92].

9.2.2.2 Racer and Other DL Reasoners

Though Protégé supports most features of the OWL family of languages through its plug-in mechanism, the OWL-based reasoner Racer does not. Racer automatically removes OWL-Full's metaclasses and ranges before reasoning begins [80, 93]. Racer can test for class, property, instance, or ontology on small sets, and verify arbitrary set-theoretic logical conditions on them (e.g., that the invariant part of an inverse of a transitive property is also transitive).

Using Racer over OWL-DL-based ontologies could offer signficant benefits to ontology construction and maintenance [94]. Because Protégé and OWL-DL represent knowledge differently and support different, not altogether overlapping ideas about classes, properties, and ranges, the conversion of Protégé-built ontologies to OWL-DL entails significant representational choices [96]. Moreover, Racer cannot reason over an entire ontology if it is very large; when Racer was applied to an OWL-DL conversion of the FMA, it could reason over the ontology only if the size and complexity of the data were reduced. The primary reduction chosen to facilitate reasoning was to limit the data to particular properties of the classes and instances. Racer then performed several logical checks:

- First, that data does not descend from two parents that have logically inconsistent properties (`Compartment_subdivision` cannot descend from both `Material_physical_anatomical_entity` and `Nonmaterial_physical_anatomical_entity`);
- Second, that the value of an instance's property is consistent with the definition of that property (e.g., the value of the property D2D_PART, which has the range `Nonmaterial_physical_anatomical_entity`, for

`Surface _ of _ wrist` is `Anatomic _ snuff _ box`, which descends from `Material _ physical _ anatomical _ entity`);

- Third, that the application of transitive relations to an object does not result in biologically inconsistent change of the object's place in the ontology, or `reclassification` (e.g., reasoning that an organ is a subclass of a cell).

In addition, Racer automatically induced the ontology's topology from a set of conditions manually defined as necessary and sufficient, revealed a number of discrepancies between the manually and automatically constructed topologies. Each check revealed inconsistencies in the ontology that had gone undetected, despite years of concentrated effort by its very seasoned builders. While the checks also revealed a number of limitations and weaknesses in ontology conversion and Racer's abilities, as would be expected, Zhang et al. persuasively demonstrate that reasoning over an ontology can improve its maintenance [94].

Pellet and FaCT++ are reasoners that implement description logics (OWL-DL and SHIQ, an early description logic, respectively) [75–79]. Pellet is intended for ontology maintenance (cleaning, format-checking, and so on), and it is offered commercially for this purpose. GRAIL, an earlier DL reasoner, supports `sanctions`—rules that scope the application of inferences to concepts in the construction of composite notions [72, 96]. Different versions of TAMBIS, a semantic mediation system for five sequence and function databases that relied on an ontology to provide a global schema, used GRAIL and FaCT++ [96]. SWRL is a rule language that supports OWL-DL, OWL-Lite, and RuleML statements [85].

9.2.2.3 The OBO-Edit Reasoner

The OBO-Edit reasoner goes a step further, by deriving new instances of the relations in ontologies [77]. It uses very simple production rules, such as genus implications, differentia implications, transitivity, and cross-product implications, to generate new instances of existing rules. It then iterates, taking the newly generated relations into account, and terminates when no new instances are found. For example, given relations `part _ of(X,Y)` and `part _ of(Y,Z)`, where `X`, `Y`, and `Z` are all instantiated anatomical structures, the OBO-Edit reasoner should produce `part _ of(X,Z)`. Though very useful, the OBO-Edit reasoner can only reason with OBO-compliant ontologies, which, for some purposes, may be structurally insufficient (see Section 9.3.2). For example, robust data that reflects phenotypic diversity among clades or individuals can be difficult to represent in OBO ontologies. Moreover, the OBO-Edit reasoner can only reason over the intensional data of the ontology, not over the observational data.

9.2.2.4 Reasoning Languages

A brief consideration of some computational technology is warranted. Many algorithms that exploit sequence, structural, or image data are written in procedural or object-oriented languages, such as C, Fortran, and C++, for which excellent libraries are available and which permit efficient memory management. In contrast,

declarative languages emphasize the efficient evaluation of logical predicates, and they originally arose from work in artificial intelligence, including natural-language processing [97, 98]. The ancestor of many is Prolog, a mature, robust, and efficient logic-programming language that implements the first-order predicate calculus as Horn clauses (FOPC; see [97]). Prolog is widely used for reasoning systems, and several Prolog compilers can natively call code written in C or Fortran for numerically intensive computations [99]. Several languages that restrict or extend Prolog in various ways, and can be used to reason with ontologies, have appeared in the last few years. For example, *Flora*-2 is an object-oriented knowledge-base language that relies on a logical inference engine written in XSB, itself a descendant of Prolog that implements several extensions of the FOPC over database tables [100–102]. F-OWL is a very promising inference engine [103]. Written in *Flora*-2, it is designed to work with ontologies expressed in one of the OWL variants. F-OWL offers more logical power than languages in the OWL group [104]. `Jess`, written in Java for the development of expert systems, supports many different types of reasoning over Java objects, including rules [92]. Finally, there have been several attempts to develop languages that strike different compromises between expressivity and decidability, or that can reason over information in the Semantic Web, or both [4, 85, 105, 106]. For example, TRIPLE uses layers of Horn clauses and description logics to express rules and permit reasoning over RDF and DAML+OIL models [106]. Implementation is in Prolog or XSB.

Several less logically powerful languages have been designed to express and reason with ontologies. The archetype of this group is OWL and its ancestors RDF and DAML+OIL [82, 83, 107, 108]. All are tripartite languages, in the sense that they allow each operator two arguments of the form `subject verb object`. OWL comes in three flavors with increasing logical power (OWL-Lite, OWL-DL, and OWL-Full); nearly all work has been done with the first two, which limit their logic to the set operations of the FOPC to preserve decidability. These and other limitations sharply restrict the kinds of inferences these languages can draw, compared to Prolog. For example, the difference between OWL-Full and OWL-DL is that the former permits classes to be instances of other classes (metaclasses), and an element can be an instance, class, or property, without separating these into disjoint sets. So OWL-DL reasoning supports consistency checking and classification (subsumption), but the use of metaclasses in OWL-Full causes rejection of the ontology. OWL-DL implements `description logic`, which characterizes concepts by enumerating members of sets that can be classified as instances of those concepts [109]. However, the name *description logic* is somewhat of a misnomer; description logics do not describe entities or phenomena in the sense of specifying the properties that something must fulfill in order to be a member of a set; rather, they enumerate the members of a particular set. Consider the following example from plant developmental anatomy. The set of inflorescences (each inflorescence is a cluster of flowers on a stem or branch [110, 111]), might be defined by biological rules, such as "develops from infloresence meristem," "has floral parts," "can be monecious," and "can be dioecious." These terms might be associated with a particular instance of a structure and used to demonstrate if that structure was an infloresence. (Of course, each of these rules would require computational definitions of their ideas, such as inflorescence meristem, development, floral parts, and

so on.) In constrast, a description logic might say, "A tassel is an inflorescence," "a tassel is the male inflorescence in maize," "structure *X* is a maize tassel." In the case of biological rules, a reasoner would have to be able to analyze an instance of a structure and determine if the rules were true for that instance. In the case of description logics, the reasoner need only be able to execute `modus ponens`: "If a maize tassel is an inflorescence, and *X* is a maize tassel, then *X* is an inflorescence" (and similarly for the inflorescence's gender) [112].

9.2.3 Anatomy as a New Frontier for Biological Reasoners

9.2.3.1 Rationale

There are several reasons why anatomy from diverse phylogenetic groups is such a good test case for biological reasoning. First, anatomical information provides a locating framework for the description of phenotypes in many organisms, whether these are developmental mutants in a single species, the population variation exhibited by individual members of a species, or the anatomical changes used to differentiate among species. Second, taxa differ in their anatomical descriptions and, for large phylogenetic distances, even in their organizing principles. Thus, expressive representation and reasoning about the anatomy of wild-type and mutant individuals and disparate taxa force one to address biological diversity. Third, reasoning about anatomy requires the logical definition and software implementation of ideas, some of which have been defined incongruously heretofore, or have so far escaped the attention of computational biologists. Fourth, many anatomical ontologies now exist, allowing one to consider their various approaches to knowledge representation and reasoning. Finally, primary anatomical data in the form of digital images are available for some taxa, providing a platform, albeit incomplete, for developing and testing reasoners.

Four key principles must be taken into account when developing anatomical ontologies with the intention to reason over them. First, as with any knowledge representation, the ontology must be appropriately modularized; that is, it must be organized in such a way that it both represents biological reality and allows for the delineation of knowledge at many levels of granularity. Second, the ontology must be organized in an appropriate hierarchy, so that knowledge can be inherited from one or more parent classes through a logically consistent flow. Third, if the intention is to reason over multiple anatomy ontologies, there must be a common or at the very least, transparent, top-level hierarchy among the ontologies that will allow linkage or mapping among them. And finally, given that a main intention of ontologies is to allow for semantic integration of multiple data sources, the ontology must be consistent in the way it represents data and avoids duplication.

9.2.3.2 The Digital Age of Morphology

Many systematists now use *Morphbank* to store their two-dimensional (2D) high-resolution images and comment on specimens used in identifying and documenting species nomenclature [113]. *MorphoBank*, a comparable resource, offers storage for 2D images, with its primary focus on creating a collaborative Web workspace for

phylogenetic studies [114]. It provides online tools for the development and sharing of character matrices (a character is a phenotype that varies within or among species), the association of elements of those matrices with images, and the display of labeled images. The construction of matrices, labeling of character names, association of characters with images, homology inferences, addition of provenance metadata, and access control are all done manually by the contributing scientists. The *DigiMorph* digital library is a large collection of industrial CT scans of various taxa, and the *MorphologyNet* digital library is an online community repository for three-dimensional (3D) images of anatomy that have been generated by any method (e.g., micro-CT, MRI, histology, and so on; see [115, 116]). The success of these projects clearly shows that modern approaches to studying biodiversity require digital images and that the morphology community has embraced their use. This paradigm shift and the resulting explosion of digital images have opened the doors to the development of automated reasoning over primary anatomical data.

9.2.3.3 Documenting Phenotypes

Geneticists commonly photograph organisms or their structures as forms of primary data, such as images of mutant plants, in situ hybridizations of tissue sections, or cytogenetics [117–119]. A large number of phenotype images are available via MaizeGDB, and efforts to develop algorithms that automatically recognize and classify images of phenotypes are underway [37, 120]. Nonetheless, to the best of our knowledge, there are no 3D-image datasets of the sort made in human and veterinary clinical practices or those that animal taxonomists now collect increasingly routinely.

9.2.3.4 Anatomical Ontologies

Many OBO Foundry ontologies represent anatomical knowledge. Several ontologies represent a single model species, such as the *C. elegans* Gross Anatomy Ontology, the Mouse Adult Gross Anatomy Ontology, and the human Foundational Model of Anatomy [71, 121, 122]. Others reflect diversity in taxonomic groups, including the Teleost Anatomy and Development Ontology (TAO), the Amphibian Anatomical Ontology (AAO), and the Plant Structure Ontology (PSO) of the Plant Ontology [123–127]. These ontologies often contain robust sets of phenotypic data for many species; for example, the AAO, an OBO-based ontology, includes anatomical and developmental information and phenotype annotations, including species differentia essential to biodiversity studies [124, 128]. The Common Anatomy Reference Ontology (CARO) and the Über Anatomy Ontology (UBERON) are upper-level amalgamating ontologies that aim to map structures from one species to another [129, 130]. An alternative to manually constructing reference ontologies is to automatically align the individual ontologies. Several experiments along these lines demonstrate that it is possible to align multiple anatomical ontologies for a single organism (humans) and across mammalian species (humans and mice), though so far, the alignments are not complete [95, 131, 132]. In these experiments, both terms and subgraphs of relationships and terms were used to align ontolo-

gies directly to each other or with respect to a third ontology that served as an intermediating reference.

9.2.3.5 Algorithms and Tools for Systematically Linking or Aligning Anatomy Ontologies

Several ontological resources (e.g., the UMLS Semantic Network and the Foundational Model of Anatomy) currently are being utilized to support biological text mining and to assist in entity-recognition tasks and relation-extraction tasks [133, 134]. The use of information retrieval and extraction tools to automate the process of building an ontology, however, is not yet a common practice; see [135–137] for examples of research in that direction.

Similarly, fully automated alignment of ontologies is not yet a well-established field. The work in [138] describes several approaches to ontology merging and the calculation of differences, functionality that has been implemented in a limited form in PROMPT and OntoMerge [139, 140]. Additionally, the authors of [138] introduce a practical approach for ontology merging and the calculation of difference. RDBOM, a relational database ontology-management system, also provides for ontology merging and difference calculation [141]; however, the RDBOM implementation is based on a finite-state automaton ontology model called an *ontology abstract machine* [142], while the work in [138] is based on description logic (DL).

A software tool for the partially automating alignment of OBO ontologies is available (as described in [143]); however, here alignment is not semantically equivalent to the functionality considered in [138] and [142]. Rather, it refers to the linking of ontologies. A Perl script allows the user to specify cross-reference (OBO `xref`) linkage information in the form of a text prefix. That information then is used to link terms in the designated ontologies that match those text patterns; the actual linkage is achieved by automatically adding an `intersection_of` entry (which contains the cross-reference information) for each matching term in the OBO file. The OBO-Edit reasoner can be used subsequently to analyze these links for consistency in terms of `is_a` relationships (i.e., to check if two linked terms refer to different kinds of entities, based on the corresponding `is_a` relations that are defined in the original ontologies).

9.3 Biological Ontologies Today

9.3.1 Current Practices

The current practices for building biological ontologies are rooted in the way ontologies were initially developed and used in model organism databases. Indeed, the persistence of these practices nearly a decade later, and their widespread adoption by biological ontology builders, are a tribute to the perspicacity and tenacity of the original developers and their colleagues in the Gene Ontology Consortium (now the OBO Foundry) [62]. The Gene Ontology, which is the model for most of the biological ontologies in production use today, was developed very rapidly, under

enormous pressure, for the annotation of the about-to-be-published *Drosophila melanogaster* genomic sequence [144]. It was so successful that it was only natural to hope it would become a tool in unifying biological language, so that databases could become semantically interoperable. This is not unreasonable; the Edinburgh Mouse Atlas and the Jackson Laboratory's Mouse Database showed that databases built with the same ontology would indeed be semantically interoperable [145, 146].

The rapid adoption of the Gene Ontology by the mouse, *Arabidopsis thaliana,* and the zebrafish database communities clearly indicated its value and greatly expanded its coverage by providing additional seasoned curators to join the open process of ontology construction. Each extant ontology is the result of taxing effort and careful reasoning over divergent views to arrive at a reasonable terminology. Perhaps most importantly, the growth of the Gene Ontology has made many more people "representation conscious," or aware that there are choices to make in representing biological information, by stimulating similar efforts for other organisms and categories of ideas. Several groups have described or recommended practices in ontology construction [7, 67, 131].

The fundamental Gene Ontology model of a forest of related ontologies, each a directed acyclic graph and all expressed in a tripartite language, persists today because of its success as an annotation tool. Moreover, the Gene Ontology and its relatives have been used for many purposes never originally anticipated, for example, in automated text processing [7, 147, 148]. When pressed for uses its designers never envisaged, there can be problems, but we know of no model or software system that is perfectly forward compatible.

9.3.2 Structural Issues That Limit Reasoning

Given their ubiquity and computational raison d'être, it seems natural to use ontologies to automatically discover knowledge by emulating biological reasoning. Their usefulness in information organization and in the sharing of information among data resources has been amply demonstrated by the adoption of ontologies by the biological databases mentioned above and many others [2]. In some cases, for example, in TAMBIS, an ontology is used to form a de facto global schema and serve queries over multiple databases [96]. In many cases, ontological terminology forms the semantic content of automated queries from one portal or database to another or guides the incorporation of information from multiple sources into a unified presentation (e.g., the databases of the Entrez collection of NCBI and others that connect to them). In still other cases, ontologies supply semantic mappings in mediation schemes, especially over the Semantic Web [24, 25, 149]. Moreover, Chapters 3–8 and 10 in this book amply illustrate the use of ontologies by algorithms to retrieve, annotate, and filter data.

Reasoning about anatomy reveals some structural limitations that particularly plague anatomical ontologies and that must be addressed before they can be used effectively for automated knowledge discovery. By *structural*, we mean the fundamental constraints that some ontology-development tools, such as OBO-Edit, impose on the representation of knowledge, not the terminology, relations, or topology of any particular ontology. For example, early versions of OBO-Edit did not

include provisions for adding instances or properties, nor did they allow node-to-node linking beyond the hierarchical is_a and part_of. We emphasize that our interest is the fundamental structural issues of the typical biological ontology, rather than the terminological or topological particulars of specific ontologies or spatial representations [87, 150, 151].

9.3.2.1 The Imperative of No Ambiguity

Perhaps the most overriding technical imperative in ontology construction is to avoid ambiguity in the semantics of its terms. Biological language is often creatively stretched to capture the intrinsic variations of phenomena, but this introduces multiple, usually related, meanings into the set of definitions users associate with the terms [152]. Operatively, the semantics of many biological terms depends on their biological context, but most software is built to be as context-free as possible. The computational need to avoid ambiguity often prevents one from representing concepts in an ontology in the same manner as they are used in biological practice.

This imperative has produced two significant computational constraints on contemporary biological ontologies. First, every term in the ontology is required to be unique. Whatever semantic portfolio that term may carry in the minds of biologists, it will be represented only once. Second, *multiple inheritance* (e.g., a term inheriting semantic or other attributes from more than one term) is discouraged and is not considered a best practice [82]. This preference can be awkward; many anatomical entities are nearly always determined by other structures that are adjacent to them or that preceed them temporally or mechanistically [153]. Thus, contemporary ontologies, such as the Gene Ontology, use it cautiously [66].

9.3.3 A Biological Example: The Maize Tassel

In principle, a reasoner would recognize anatomical, phylogenetic, spatial, or developmental relationships among phenotypes present in individuals of one or more species. How well do current ontological practices permit representation of this essential information? To answer this question, we show the representational problems posed by the anatomy of the maize male inflorescence, or tassel.

9.3.3.1 Anatomical Modularity

Maize is built from a series of repeated modules that differentiate into organs throughout the plant's life [153, 154]. Sets of smaller modules nest inside the larger ones to build the organs and their substructures, beginning with the individual gametes of the pollen grains and progressing to the entire tassel. Figure 9.1 illustrates this modularity.

One can divide the tassel into large and small modules; a wild-type tassel can have more than 10 large modules and 100 small ones. The large modules are the "limbs" of the tassel: an axial, central spike, and its lateral branches [155, 156]. As in the tassel in Figure 9.1(a), this spike-branch structure can be repeated in place of the lowest branches, with the secondary axis and its branches forming where a simple lateral branch would be. Each large module is covered with small spikelet

(a) (b)

Figure 9.1 The nested, modular structure of the maize tassel. (a) shows a tassel with its large modules (central and branch axes and lateral branches) covered with pairs of glumes (the small modules). The arrows mark two subsidiary axes with their own central axes and lateral branches. Many glumes have opened, revealing the florets and their pairs of anthers (triangle). (b) is a close-up showing the alternate arrangement of spikelets and their pairs of glumes. The arrow indicates the group of three stamens, each with its pair of anthers, that are the reproductive parts of the male floret.

modules, arranged alternately as shown in Figure 9.1(b). Each spikelet bears two glumes, again alternating across an imaginary axis. Each glume encloses a floret with two clamshelllike plates, the palæ and lemma, that shelter the three stamens inside until their pairs of anthers dehisce and shed pollen.

Changes in Anatomical Parameters During Development Phyllotaxically, maize leaves alternate along the central axis of the plant's stem, and their placement along that axis can be described as a helix:

$$h(\theta) = r\cos(\theta)\boldsymbol{i} + r\sin(\theta)\boldsymbol{j} + p(\theta)\theta\boldsymbol{k}$$

where r is the radius, $p(\theta)$ is the (changing) pitch function of the helix, θ is its curvature (loosely, the angular rate at which the helix turns), and $\boldsymbol{i}$, $\boldsymbol{j}$, and $\boldsymbol{k}$ are the usual 3D unit vectors. (While the developmental context makes it particularly tempting to also interpret θ as time, for the moment we eschew that connotation for lack of direct evidence.) For simplicity, we call the interval along the helix between two successive modules on the helix, the *arc interval;* and the angle subtended by two successive modules, the *angular interval*.

This helical pattern of leaf placement on the stem continues on the tassel's central spike, secondary axes, and branches. As the vegetative helix switches to an inflorescence helix and proceeds up the tassel, two things can happen. First, the frequency of the placement of branches and spikelets can increase, progressively decreasing the arc interval and the angular interval, and breaking the strict 180° phasing of the leaves. In a limited sample of approximately 75 wild-type tassels from four different lines of maize, this compression always occurs. Second, a second helix can appear, slightly out of phase with the first, resulting in pairs of branches very close to each other, on the same side of the axis. The appearance of the second helix depends on the genetic background. At present it is unclear whether the angle subtended by the branch and the axis (the *branch angle*) is determined independently of the helix parameters.

Geneticists commonly measure several parameters in describing tassel morphology [157, 158]. Most of these phenotypes are direct results of the parameters described and others are global properties of the tassel, while one is apparently independent. The measured phenotypes include the total length of the tassel, the average length of three branches drawn from different locations in the tassel, the length of the central spike, and the length of the branching zone; the number of branches; the number of spikelet pairs in the densest regions of the central spike and lowermost primary branch; the average angle between the central axis and the branches, projected onto a plane parallel to the length of the tassel; and the tassel's dry weight[1].

9.3.4 Representational Issues

With the tassel's anatomy clear, we now can consider how the current ontological practice may obfuscate an accurate representation. We suggest there are three problems surrounding the representation of anatomy in ontologies and the consequent problems they present for reasoning.

1. The standard method of storing a dried, flattened tassel in a paper bag for later measurement makes it difficult to measure θ directly, and collecting the data to estimate the pitch function p(θ) would involve too many manual measurements to be practical for thousands of tassels. Quantitative estimates of these patterns for large populations, which would facilitate identification of mutants in their production, will require improved technology.

The Multiplicative Crisis The first problem arises from the duplication of anatomical modules, or serial homology, that is demonstrated by many organisms. Since each term in an ontology must be unique, every anatomically identical instance of a module must have its own name. Thus, in an effort to distinguish each individual anatomical duplication, an ontology may have many names for the same structure, and artificial distinctions among identical modules may misrepresent an organism's anatomy when viewed through the ontology.

This gap between terminology and modularity has been called the *multiplicative crisis* by Elizabeth Kellogg and Lincoln Stein [159]. The degree of the crisis varies by taxonomic group. Since animals have few identical modules, animal anatomy ontologies customarily treat each anatomical structure as a distinct entity, avoiding anatomical distortion by simply ignoring modularity or naming each module separately. In contrast, the extensive modularity of plants makes it infeasible to give each module its own name, as our tassel example shows. In practice, the PSO does not represent every module, choosing instead to represent characteristic parts of a few modules. This decision is perfectly sensible when the goal is to have a basic list of anatomical structures, but becomes awkward when the goal shifts to representing the plant's anatomy for reasoning. Figure 9.2 shows a portion of the tassel's anatomy roughly corresponding to the large and small modules of our description, as represented by the `part_of` relations in the PSO. The figure was produced from data extracted by manual inspection of the flat-file ontology [125].

Structures that are the same, in the context of the tassel, but named differently in the ontology to distinguish between the general type of a structure and its organ-specific versions, share the same gray color. Thus, the left portion of the figure shows the relations among the more general terms, such as spikelet, glume, floret, gynoecium, and androecium, and the right portion shows the tassel-specific versions (there are parallel ear-specific versions for most of these in the PSO). Palæ, lemma, stamen, and anther are ambiguous, in the sense that it is not clear whether general or tassel-specific versions are meant; only a maize biologist would know that anthers are found only on tassels because corn is monecious.

We suggest that there are five challenges in representing modularity with current ontological practice:

1. *Representing Module Structure*—The first challenge is that there is no device to clearly represent the anatomical modules as modules. To indicate that a particular structure occurs in the tassel, one must name both the type of structure, for example, the floret, and that structure in the tassel, for example, the tassel floret. As Figure 9.1 shows, however, a tassel has many florets. By looking at the ontology and a tassel, one can recognize the portions of the small module implied by the tassel floret and its children, but a method for explicit representation is missing.
2. *Representing Number*—For all tassels, one of the variables in describing their phenotypes is *number*. All wild-type and nearly all mutants of maize have many, many more than one branch and floret, and the phenotypes of some mutants, such as *ramosa1*, are distinctive, because the numbers of their branches and florets are far greater than usual [156]. Moreover, the individual tassels of a particular genotype will exhibit a range of numbers

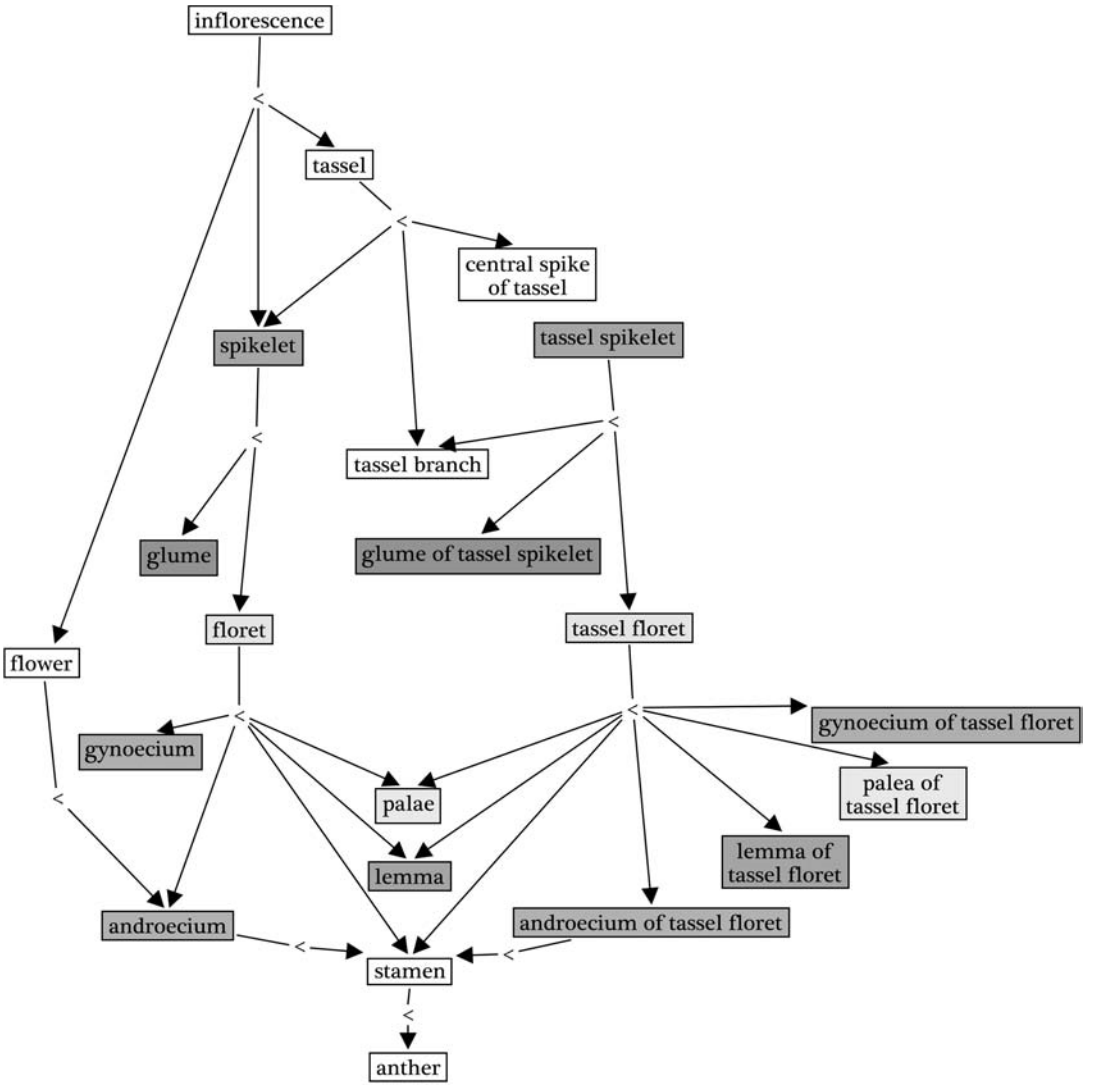


Figure 9.2 Tassel modules as represented by the PSO's part_of relations. Structures that are identical in the tassel but named differently in the ontology share the same color. Notice that the ontology represents only one instance of each module, but with the exceptions of the tassel and its central spike, each entity is present many times in any real tassel. The `sensu` terms have been omitted for clarity.

of branches, florets, and florets on branches. Yet the ontology says that a tassel has one central spike, one branch, and one floret. How one would say how many modules a particular tassel has is unclear.

3. *Representing Positional Information*—In wild-type and many mutant tassels, the florets in the middle of the central spike develop first, with development proceeding in waves up and down the spike as the florets on the branches mature (in approximately the same wave pattern). To describe this process, one must distinguish the central spike from the branch florets, the different positions of florets on each limb, and the positions of the limbs

themselves. In principle, some of this information could be conveyed by permitting an instance of a small module to inherit some of its positional information from the subsuming large module and the rest by referring to the positions of its adjacent modules. However, this would result in multiple inheritance, a discouraged practice and one most ontologies try to avoid.

4. *Representing Development*—The ontological representation of the developmental progression of structures is limited by the unease with multiple inheritance. Figure 9.3 shows a manually constructed extract, from the PSO, that involves a portion of tassel development.

   The `short tassel branch meristem` is a `part_of` the `central spike` of the tassel, the `long lateral tassel branch meristem`, and the `long lateral tassel` branch. However, in the mature tassel, growth and differentiation of the meristems have ceased. A short tassel branch is not part of a long tassel branch, and the central spike and branches are differentiated structures. If multiple inheritance were freely used, a reasoner could use it to discover these biological inconsistencies. For example, these biological collisions would be signaled by the short tassel branch meristem simultaneously inheriting properties, such as adult structure, developmental precursor, apoptosis, and relative position, from its ontological parents.
5. *Representing Properties*—Finally, there is no mechanism to describe the properties of tassels, either local to a particular region or globally for the entire organ. The tassel's phyllotaxic and phenotypic parameters are morphological properties just as central to the tassel's structure as the modules, glumes, and florets. Similarly, number, position, and temporal information are all directly relevant to changes in tassel morphology caused by genetic, developmental, and environmental processes. Even if the terminology and representations were supplied by other ontologies, such as those for traits or phenotypes, the structural limitations of the ontologies that preclude accurate representation of the anatomy would prevent their use in describing morphological changes.

9.3.4.1 Term Synthesis

To distinguish connotations, modules, and locations in ontologies, often novel terms are synthesized. Nodes of the same color in Figure 9.4 illustrate combinatorial neologizing, for example,

$$\{\texttt{lower} \vee \texttt{upper}\} \wedge \{\texttt{glume} \vee \texttt{lemma} \vee \texttt{palae} \vee \texttt{floret} \vee \texttt{androeium} \vee \texttt{gynoecium}\} \wedge \{\texttt{sessile} \vee \texttt{pedicellate}\} \wedge \{\texttt{spikelet}\}.$$

The extent to which biologists who are not curators use such combinatorial terms is unknown, but experience suggests such usage is essentially nil. Ogren et al. have provided detailed descriptions of the resulting term compositionality and the issues this raises for text-mining applications [160, 161]. While their examples are grammatically more complex than ours (e.g., "positive regulation of cell migra-

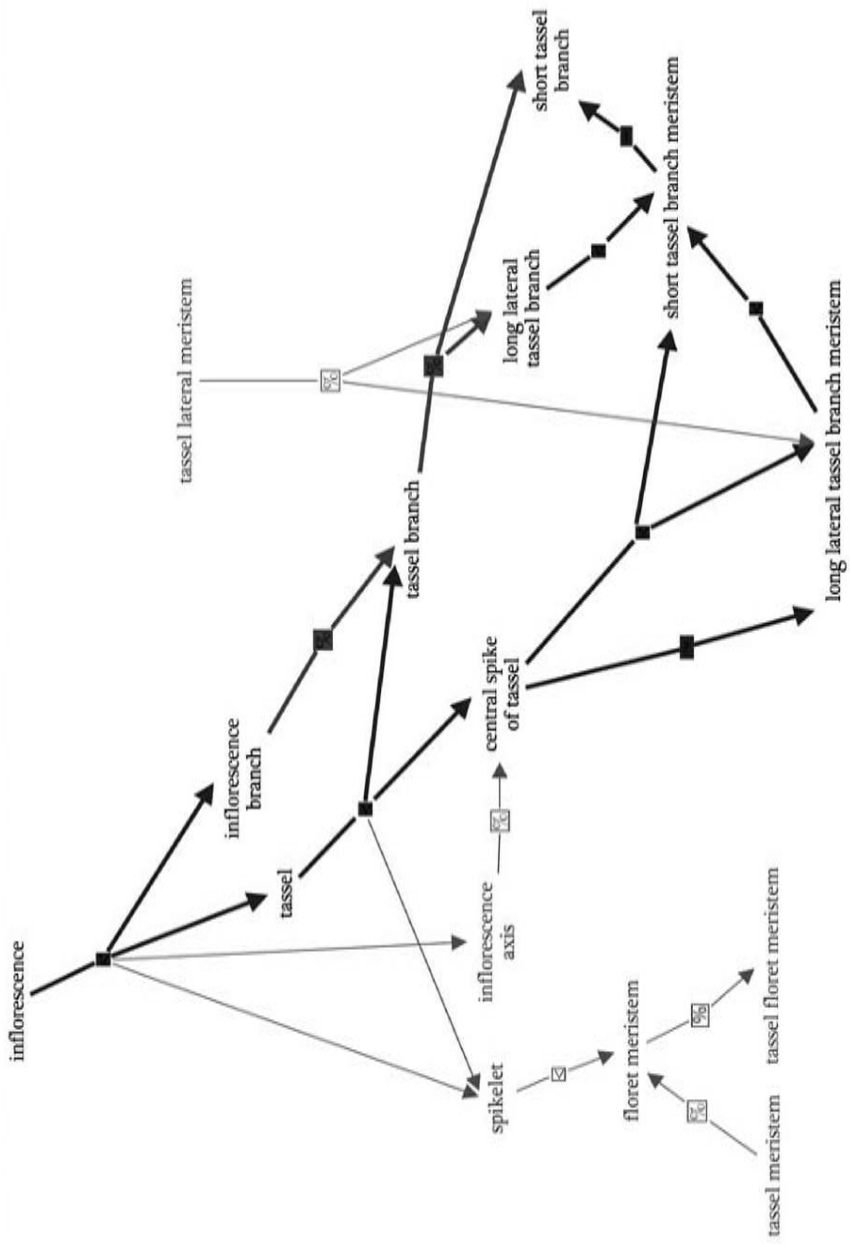


Figure 9.3 A part of tassel development, as represented by the PSO. Developmentally important relations are in black: `part_of` (<), `is_a` (%) and `develops_from` (~); their ontology context is given by the gray relations. The representation does not distinguish temporal or differential relations among the structures, producing biological inconsistencies.

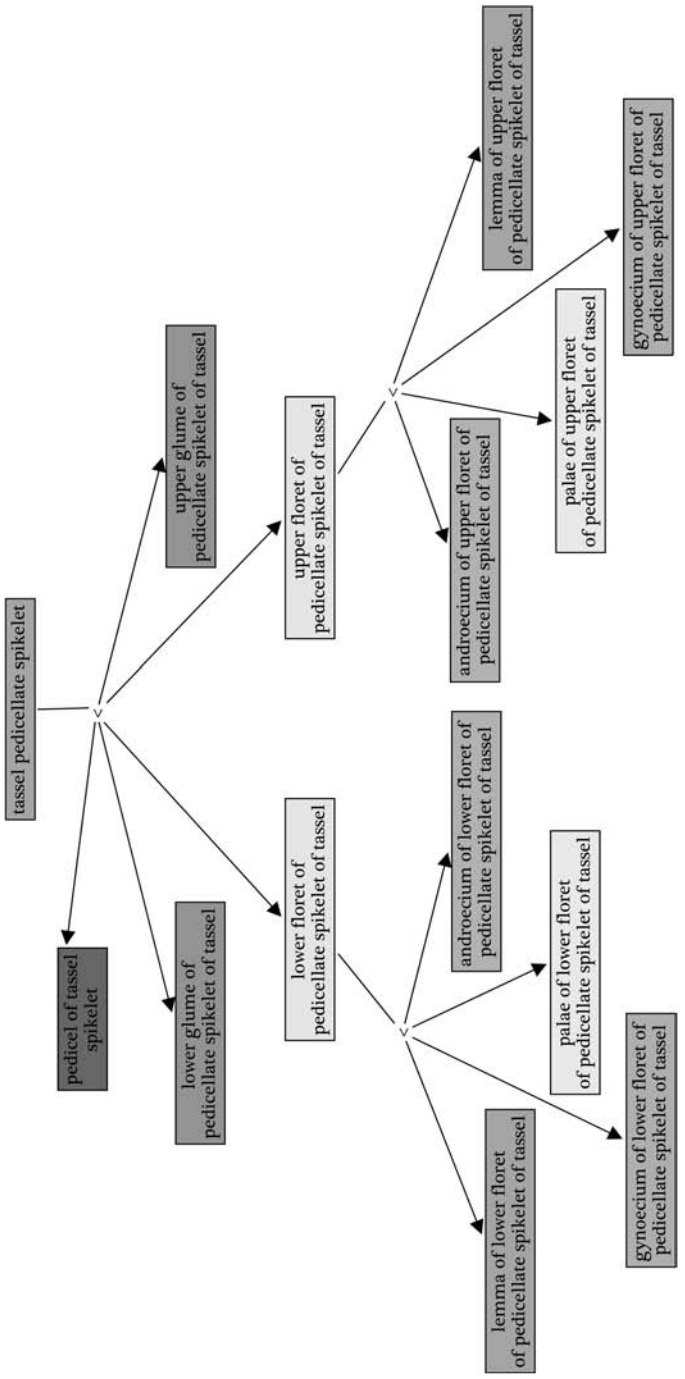


Figure 9.4 Enforced neologizing in an attempt to record the position of anatomical modules. Identical parts differing only in their relative position in the spikelet pair are the same shade of gray. The pedicel distinguishes pedicellate and sessile spikelets.

tion"), the types of composition they describe for the Gene Ontology also occur in the PSO.

This synthesis of "artificial" terms often occurs, because most ontologies rely on terms that approximately describe a wild-type member of a species. But biodiversity, genetics, and developmental biology compare multiple specimens, looking for subtle differences among them. Upper-level ontologies, such as CARO and UBERON, seek to provide a unifying framework for mapping among ontologies of individual species, but sacrifice precision to do so [129, 130]. In addition to creating artificial terms, they often must rely on constructions based on analogy, rather than on homology (UBERON); or they must prune data, such as robust taxon-associated phenotype annotations or character codings.

One way in which terms are synthesized is through the infix operator `sensu`. Depending on the direction in which it is read, *X* `sensu` *Y* performs different operations. As a postfix operator for *X*, it *differentiates* types of *X* according to the information in *Y*. As a prefix operator for *Y*, it *amalgamates* all the various types of *Y* into a subsuming entity *X*: an approximate synonymy. Ontologies often use `sensu` to indirectly denote taxon-specific anatomy. Thus, `leg sensu Drosophila`, `anther sensu Poaceae`, leaving the computer to somehow discern the anatomy [125–127].

9.3.4.2 Tripartite Languages

There has been some discussion as to whether tripartite languages permit accurate representation of the biology and complex ideas in general [104, 162]. At least two counter-arguments have been made: first, that a proof by C. S. Peirce [163] exists that shows that languages of this form are sufficient to represent anything; and second, that expression of more complex notions simply entails more predicates, each successively unpacking notions that the previous ones had subsumed into a more general notion [164].

A look at the helical equation for the tassel's phyllotaxy might prompt some rethinking. The combined equation has one variable ($h(\theta)$) and two parameters (r, θ). Rewriting it into its three component equations doesn't help very much, because all three component equations must be simultaneously true—the essence of parametric equations. Perhaps judicious lumping might resolve the problem by letting the right-hand side be a single entity and then successively lumping and splitting each term. It is not clear how software for numerically solving equations could solve such a decomposition per se, or whether the effort to write one to handle ontological representations would be justified, given that many excellent numerical languages and packages for mathematical and statistical computations already exist [165–170].

9.4 Facilitating Reasoning About Anatomy

The problems described in Section 9.3.2 illustrate the challenges for automated reasoners that would exploit ontologies for studying variations within and among

species. If all that ontologies could offer reasoners were structural relationships among ideas, one might be tempted to use something else.

However, we believe discarding contemporary ontologies for something more structurally pure would be a mistake. Contemporary biological ontologies are packed with nouns denoting parts, processes, and molecules, and these often are, or are derived from, official biological nomenclature. For reasoners, the terminology is the proverbial baby. So we suggest combining the terminology in ontologies with different software, representational approaches, or computational techniques for reasoning.

We now consider add-ons or modifications to anatomical ontologies that will help in the development of robust reasoning systems for discovering knowledge about biodiversity, genetics, and developmental biology. These either circumvent the restrictions of anatomical ontologies, by loosely or tightly coupling the ontology terms to independent data sources representing biodiversity information, or they abandon best ontology practices when developing anatomical ontologies.

9.4.1 Link Different Kinds of Knowledge

One way to reconcile the conflict between good ontology practices and incorporating species and phenotypic diversity into anatomy ontologies is the approach taken by the Phenoscape group, which uses software tools to loosely couple independent data sources representing different kinds of knowledge about biodiversity, such as anatomy, taxonomy, and species differentia [171]. Semantic connections between the ontology's concepts and trait information from PATO are represented as OWL `Entity Quality` statements, which are created manually by curators, using software such as Phenote [172–174]. The `Entity` is essentially the anatomical structure or process (anything identified with an OBO-family identifier, such as one from the GO), and the `Quality` is a descriptor drawn from PATO. Some of the structural problems described in Section 9.3.2 could be managed by this syntax, which allows for numbering in some situations, tags for conditions that trigger phenotypic expression, and temporal qualifiers [174]. For others, extensions would be needed.

9.4.2 Layer on Top of the Ontology

An alternative approach is to tightly couple ontologies and other data sources representing biodiversity and developmental data into a single, inclusive system. Layering representations and computations in other languages and systems, such as C, Prolog, and relational database management systems, on top of ontologies would avoid many of the structural limitations described in Section 9.3.2. Layering has the advantages of being compatible with current ontology best practices and facilitating reasoning, but it entails mapping some representations in the ontology to the other components to permit them to exchange data; resolving syntactic inconsistencies among the different components; increasing the complexity of relationships among the components; and decreasing ease of expression. One consequence of this approach would be neologizing to construct terms that combine properties, such as modularity, number, position, and so on, when translating to an OBO representation. Much of this artifical terminology could be reduced simply by translating the

ontology into OWL-DL, which has ample support for properties and is decidable. Losing ease of expression might well prove to be the greatest problem in the long run, especially as the biological ideas and their corresponding computations become more complex.

9.4.3 Change the Representation

Some of the structural gaps in Section 9.3.2 can be managed by linking and layering approaches, but an alternative would be to modify current biological ontology-building practices by changing the representational techniques used. For example, switching to OWL-DL en lieu d'OBO would facilitate adding properties, including numerical ones, representing biodiversity and anatomical and developmental knowledge to the ontology. The logical limitations of OWL-DL and Racer make them highly unlikely to support much reasoning over the observational data, even if the data could be appropriately represented in those languages. However, using them in a layered approach could increase power. The modularity, number, position, temporality, and properties of tassels described in Section 9.3.3 might be represented in OWL-Full, exploiting metaclasses, numbers, and ranges; reasoning over this representation would require something more powerful than Racer.

A different alternative direction would exploit logical rules. Contemporary ontologies assume all instances of a class are enumerated, but often one wishes to recognize whether and how well a biological datum fits a pattern or rule. For example, when applied to biodiversity and genetics, one could plausibly expect a reasoner to classify specimens according to defined criteria, recognize anatomical structures from CT data, or identify outliers for expert consideration. Logical languages, such as Prolog, *Flora*-2, and XSB, surely combined with other languages better optimized for numerical computations, would offer some possibilities. One could simply reuse the data in existing ontologies and modify their syntax to let them serve as extensional data for a reasoner. In addition, moving away from the current set of tripartite languages to a more logically powerful language could encourage reorganization of the data and ideas in the current ontologies, enhancing their usability for reasoning. Compared to a shift in languages, a syntax modification would be simpler to implement and would preserve reuse of the unmodified data for Semantic Web applications, but it would sacrifice expressivity.

Another set of alternatives lies in reasoning approaches beyond the first-order predicate calculus or its subsets that might better express the reasoning of actual biologists. Heuristics that model intuition, second-order logic, constraint-logic programming, case-based reasoning, and modal logic are just a few examples. We think it particularly important to include support for temporal reasoning, given that genetics, development, and evolution occur over time in an at least partially ordered sequence of events.

A less fundamental option would be to use multiple inheritance more freely in an ontology that is extensively augmented with properties. While this explicitly breaks current ontology best practices [175], it seems less revolutionary than the other options discussed. Along with increased power, however, multiple inheritance introduces a raft of problems in designing a system to detect and resolve inconsistencies among inherited properties in a computationally and biologically reasonable

manner [176]. Better use of namespaces and scoping within an ontology would also reduce the number of synthesized terms, and these might be part of a redesigned ontological structure that enthusiastically exploited multiple inheritance. It seems reasonable to expect that knowing the biological context of a property, perhaps for all the parental nodes, would be helpful (if complicating) in building such a resolver. There may be no uniform way to resolve such inconsistencies, but that assessment might well be too pessimistic, especially for well-defined domains.

9.5 Some Visions for the Future

Reasoners for biological ontologies are still in their infancy. Although many algorithms already reason about observational data, numerical processes are quickly becoming distributed over computational grids, and mediation technologies, such as the Semantic Web, are increasing, these resources do not automatically a biological reasoner make. In an ideal world, reasoners could discover knowledge directly from primary data, with little-to-no human intervention required. Making that goal a reality can involve any of many possible next steps in reasoner development that includes ontologies, as we outline below.

In the penultimate step towards full reasoning over observational data, ontologies could serve simply to mediate any distributed computations needed, especially for very large datasets or numerically intensive computations. An intermediate step would accelerate reasoning over observational data by using the ontologies in preliminary reasoning about a problem, reducing the computations over the observational data by this initial filtration.

Reasoning over primary data could be used to construct or check ontologies. It would be ideal if the application of a term to data was accompanied by a report that traced the human and algorithmic rationale for, and contributors to, that application. Many databases already do something similar, by automatically generating annotations about sequence features, possible functions, and protein similarity matrices from sequence data [58, 177–179]. Finally, reasoners could be used to check ontologies for logical and biological consistency, along the lines of Zhang et al. [94].

None of these intermediates are trivial; all will require significant creativity and consummate attention to detail. Converting the challenge of automated reasoning about complex biological phenomena into reality should generate significant excitement in the coming decade.

Acknowledgments

We thank Olivier Bodenreider, Ed Coe, Georgia Davis, Michael Gerau, Lawrence Hunter, Yves Lussier, Eric Neumann, Alan Rector, Mary Schaeffer, Robert Stevens, and Armani Valvo for helpful discussions. This work was supported by grants from NIH GM56529 and from the University of Missouri Research Board to T. K., and NSF DBI-0640053 to J. L. and A. M.

References

[1] Anonymous, "Ignorance Is Not Bliss. Why This Is Bad," *Nature*, Vol. 430, 2004, p. 385.

[2] Bodenreider, O., and R. Stevens, "Bio-Ontologies: Current Trends and Future Directions," *Brief. Bioinfo.*, Vol. 7, 2006, pp. 256–274.

[3] Cannata, N., et al., "A Semantic Web for Bioinformatics: Tools, Systems, Applications," *BMC Bioinfo.*, Vol. 9, 2008, p. S1.

[4] Parsia, B., et al., "CautiouslyAapproaching SWRL," Technical Report, University of Maryland, http://www.mindswap.org/papers/CautiousSWRL.pdf, 2005, last accessed May 27, 2009.

[5] Clark, T., S. Martin, and T. Liefeld, "Globably Distributed Object Identification for Biological Knowledgebases," *Brief. Bioinfo.*, Vol. 5, 2004, pp. 59–70.

[6] Ruttenberg, A., et al., "Advancing Translational Research with the Semantic Web," *Brief. Bioinfo.*, Vol. 8, 2007, p. S2.

[7] Alexopoulou, D., et al., "Terminologies for Text-Mining: An Experiment in the Lipoprotein Metabolism Domain," *BMC Bioinfo.*, Vol. 9, 2008, p. S2.

[8] Kazic, T., et al., "Prototyping Databases in Prolog," *The Practice of Prolog,* L. Sterling (ed.), Cambridge MA: MIT Press, 1990, pp. 1–29.

[9] Lander, E. S., et al., "MAPMAKER: An Interactive Computer Package for Constructing Primary Genetic Linkage Maps of Experimental and Natural Populations," *Genomics*, Vol. 1, 1987, pp. 174–181.

[10] GIMP Development Team, "GIMP. GNU Image Manipulation Program," http://www.gimp.org/gimp.org, 2001–present.

[11] Me*VisLab* Research GmbH, "Me*VisLab*. Medical Image Processing and Visualization," http://www.mevislab.de/: Me*VisLab* Research GmbH, 2007–present.

[12] Ewing, B., and P. Green, "Basecalling of Automated Sequencer Traces Using PHRED. I. Accuracy Assessment," *Genome Res.*, Vol. 8, 1998, pp. 175–185.

[13] Ewing, B., and P. Green, "Basecalling of Automated Sequencer Traces Using PHRED. II. Error Probabilities," *Genome Res.*, Vol. 8, 1998, pp. 186–194.

[14] Beurskens, P. T., et al., "DIRDIF," http://www.xtal.science.ru.nl/dirdif/software/dirdif.html: Crystallography Laboratory, University of Nijmegen, 2008–present.

[15] Terwilliger, T. C., "SOLVE/RESOLVE," http://www.solve.lanl.gov/: Los Alamos National Laboratory, 2008–present.

[16] Terwilliger, T. C., and J. Berendzen, "Automated MAD and MIR Structure Solution," *Acta Crystall. D*, Vol. 55, 1999, pp. 849–861.

[17] Rost, B., "Prediction of Protein Structure in 1D—Secondary Structure, Membrane Regions, and Solvent Accessibility," *Structural Bioinformatics,* P. E. Bourne and H. Weissig (eds.), Hoboken, NJ: Wiley-Liss, Inc., 2003, pp. 559–588.

[18] Green, P., "Phrap," http://www.phrap.org/phredphrapconsed.html: University of Washington, 1994–1999.

[19] Buchanan, B. G., and E. H. Shortliffe (eds.), *Rule-Based Expert Systems. The MYCIN Experiments of the Stanford Heuristic Programming Project*, Reading MA: Addison-Wesley Publishing Co., 1984. Also at http://www.aaaipress.org/Classic/Buchanan/buchanan.-html.

[20] Karp, P. D., *Hypothesis Formation and Qualitative Reasoning in Molecular Biology,* Ph.D. Thesis, Stanford University, CA, 1989.

[21] Chen, R. O., R. Felciano, and R. B. Altman, "RIBOWEB: Linking Structural Computations to a Knowledge Base of Published Experimental Data," *Proc. of the 5th Int. Conf. on Intelligent Systems for Molecular Biology*, *Vol. 5*, Halkidiki, Greece, June 21–25, 1997. T. Gaasterland, et al. (eds.), Menlo Park, CA: American Association for Artificial Intelligence, 1997. pp. 84–87.

[22] Pearl, J., *Causality. Models, Reasoning, and Inference*, Cambridge, U.K.: Cambridge University Press, 2000.

[23] Thiele, K., "Welcome to Lucidcentral," http://www.lucidcentral.com/: University of Queensland, Brisbane, Australia, 2008–present.

[24] Navas-Delgado, I., et al., "AMMO-Prot: Amine System Project 3D-Model Finder," *BMC Bioinfo.*, Vol. 9, 2008, p. S5.

[25] Splendiani, A., "RDFScape: Semantic Web Meets Systems Biology," *BMC Bioinfo.*, Vol. 9, 2008, p. S6.

[26] Taverna Developers, "Taverna," http://taverna.sourceforge.net/?doc=download.html: sourceforge.net, 2007, last accessed May 27, 2009.

[27] Kepler Collaboration, "Kepler Project," http://kepler-project.org: Kepler Project, 2007, last accessed May 27, 2009.

[28] Hull, D., et al., "Taverna: A Tool for Building and Running Workflows of Services," *Nucleic Acids Res.*, Vol. 34, 2006, pp. W729–W732.

[29] Neumann, E., and D. Quan, "BioDASH," Technical Report, Massachusetts Institute of Technology, http://theory.csail.mit.edu/~dquan/ismb2005-biodash.ppt, 2005, last accessed October 31, 2008.

[30] Belhajjame, K., et al.,"Automatic Annotation of Web Services Based on Workflow Definitions," *ACM Trans. Web*, Vol. 2, 2008, p. 11.

[31] Helmholtz Open Bioinformatics Technology, "HOBIT: Helmholtz Open Bioinformatics Technology," http://hobit.sourceforge.net/about.html: Helmholtz Open Bioinformatics Technology, 2008–present.

[32] BioMOBY.org, "BioMOBY.org," http://www.biomoby.org/: BioMOBY.org, 2003–present.

[33] iPlant Collaborative, "iPlant Collaborative. Empowering a New Plant Biology," http://iplantcollaborative.org/: Cold Spring Harbor Laboratory and University of Arizona, 2008–present.

[34] Wall, M. E., W. S. Hlavacek, and M. A. Savageau, "Design of Gene Circuits: Lessons from Bacteria," *Nature Rev. Genet.*, Vol. 5, 2004, pp. 34–42.

[35] Nicholson, D. E., *Metabolic Pathways*, *19th Ed.*, St. Louis, MO: Sigma Chemical Co., 1997.

[36] Michal, G., *Biochemical Pathways*, Indianapolis: Boehringer-Mannheim, 1978.

[37] MaizeGDB, "MaizeGDB," http://www.maizegdb.org/: Iowa State University, 2003.

[38] TAIR Database Staff, "The Arabidopsis Information Resource (TAIR)," http://www.arabidopsis.org/: TAIR, 2008–present.

[39] International Union of Biochemistry and Molecular Biology, *Enzyme Nomenclature. Recommendations (1992) of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology*, London: Academic Press, Inc., 1992.

[40] Moss, G. P., "International Union of Pure and Applied Chemistry, Commission on the Nomenclature of Organic Chemistry Publications List," http://www.qmul.ac.uk/iupac/: Department of Chemistry, Queen Mary and Westfield College, 1996.

[41] Moss, G. P., "International Union of Pure and Applied Chemistry, International Union of Biochemistry and Molecular Biology, IUPAC-IUBMB Joint Commission on Biochemical Nomenclature, and Nomenclature Commission of IUBMB Publications List," http://www.qmul.ac.uk/iubmb/enzyme/: Department of Chemistry, Queen Mary and Westfield College, 1996.

[42] Stein, L., S. Cartinhour, and S. McCouch, "Gramene: A Comparative Mapping Resource for Grains," http://www.gramene.org: Cold Spring Harbor Laboratory, 2001–present.

[43] Research Collaboratory for Structural Biology, "Protein Data Bank," http://www.rcsb.org/pdb/: Research Collaboratory for Structural Biology, 1995–present.

[44] National Center for Biotechnology Information, "GenBank," http://www.ncbi.nlm.nih.gov/GenBank/index.html: National Center for Biotechnology Information, 1995–present.

[45] Maddison, D. R., K. Sabine Schulz, and W. P. Maddison,"The Tree of Life Web Project," *Zootaxa*, Vol. 1668, 2007, pp. 19–40.

[46] Hogan, L. S., et al., "How Non-Native Species in Lake Erie Influence Trophic Transfer of Mercury and Lead to Top Predators," *J. Great Lakes Res.*, Vol. 33, 2007, pp. 46–61.

[47] Petchey, O. L., et al., "Size, Foraging, and Food Web Structure," *Proc. Natl. Acad. Sci. USA*, 2008, Vol. 105, pp. 4191–4196.

[48] Pearson, W. R., and D. J. Lipman, "Improved Tools for Biological Sequence comparison," *Proc. Natl. Acad. Sci. USA*, 1988,Vol. 85, pp. 444–2448.

[49] Altschul, S. F., et al., "Basic Local Alignment Search Tool," *J. Mol. Biol.*, Vol. 215, 1990, pp. 403–410.

[50] Felsenstein, J., "PHYLIP," http://evolution.genetics.washington.edu/phylip.html: University of Washington, 2008–present.

[51] Felsenstein, J., "PHYLIP—Phylogeny Inference Package (Version 3.2)," *Cladistics*, Vol. 5, 1989, pp. 164–166.

[52] Snir, S., T. Warnow, and S. Rao, "Short Quartet Puzzling: A New Quartet-Based Phylogeny Reconstruction Algorithm," *J. Comp. Biol.*, Vol. 15, 2008, pp. 91–103.

[53] Saitou, N., and M. Nei, "The Neighbor-Joining Method: A New Method for Reconstructing Phylogenetic Trees," *Mol. Bio. Evol.*, Vol. 4, 1987, pp. 406–425.

[54] Holm, L., and C. Sander, "Protein Structure Comparison by Alignment of Distance Matrices," *J. Mol. Biol.*, Vol. 233, 1993, pp. 123–138.

[55] Bailey, T. L., et al., "MEME: Discovering and Analyzing DNA and Protein Sequence Motifs," *Nucleic Acids Res.*, Vol. 34, 2006, pp. W369–W373.

[56] Krogh, A., et al., "Hidden Markov Models in Computational Biology. Applications to Protein Modeling," *J. Mol. Biol.*, Vol. 235, 1994, pp. 1501–1531.

[57] Bairoch, A., "PROSITE: A Dictionary of Sites and Patterns in Proteins," *Nucleic Acids Res.*, Vol. 19, 1991, pp. 2241–2245.

[58] Henikoff, S., and J. G. Henikoff, "Automated Assembly of Protein Blocks for Database Searching," *Nucleic Acids Res.*, Vol. 19, 1991, pp. 6565–6572.

[59] Henikoff, S. and J. G. Henikoff, "Amino Acid Substitution Matrices from Protein Blocks," *Proc. Natl. Acad. Sci. USA*, 1992, Vol. 89, pp. 10915–10919.

[60] Lyons, E., and M. Freeling, "How to Usefully Compare Homologous Plant Genes and Chromosomes to DNA Sequence," *Plant J.*, Vol. 53, 2008, pp. 661–673.

[61] Swofford, D. L., "PAUP* Phylogenetic Analysis Using Parsimony (*and Other Methods)," http://paup.csit.fsu.edu/: Sinauer Associates, Inc., 2008–present.

[62] Ashburner, M., et al., "Gene Ontology: Tool for the Unification of Biology," *Nature Genet.*, Vol. 25, 2000, pp. 25–29.

[63] U.S. National Library of Medicine, "Unified Medical Language System (UMLS) Documentation," http://www.nlm.nih.gov/research/umls: U. S. National Library of Medicine, 2004–present.

[64] International Health Terminology Standards Development Organization, "*SNOMED CT,*" http://www.ihtsdo.org/snomed-ct/: International Health Terminology Standards Development Organization, 2008–present.

[65] Schulz, S., et al., "SNOMED Reaching Its Adolescence," *Intern. J. Med. Info.*, Vol. 78, 2008, pp. 586-594.

[66] Gene Ontology Consortium, "Gene Ontology Consortium," http://www.geneontology.org/: Gene Ontology Consortium, 2003.

[67] National Center for Biomedical Ontology, "OBO: Open Biomedical Ontologies," http://obo.sourceforge.net/: National Center for Biomedical Ontology, 2005.

[68] Plant Ontology Consortium, "Plant Ontology," http://www.plantontology.org/: Plant Ontology Consortium, 2003.

[69] Library of Congress, "Library of Congress Classification Outline," http://www.loc.gov/catdir/cpso/lcco/: Library of Congress, 2008–present.

[70] OCLC, "Dewey Services DDC 22 Print," http://www.oclc.org/dewey/versions/ddc22print/intro.pdf: OCLC, 2008–present.

[71] Rosse, C., et al., "Foundational Model of Anatomy," http://sig.biostr.washington.edu/projects/#FMA: University of Washington, 2002–present.

[72] Baker, P. G., et al., "An Ontology for Bioinformatics Applications," *Bioinformatics*, Vol. 15, 1999, pp. 510–520.

[73] Schulze-Kremer, S., "Ontologies for Molecular Biology," *Pacific Symposium on Biocomputing '98,* R. B. Altman, et al. (eds.), Singapore: World Scientific Publishing Co., 1998, pp. 693–704.

[74] Haarslev, V., and R. Möller, "RACER System Description," *Automated Reasoning. 1st Int. Joint Conf., IJCAR 2001,* Siena, Italy, June 2001, *Proceedings,* R. Goré, A. Leitsch, and T. Nipkow (eds.), No. 2083; *Lec. Notes Comp. Sci,*. Berlin: Springer Verlag, 2001, pp. 701–706.

[75] Tsarkov, D. and I. Horrocks, "FaCT++ Description Logic Reasoner: System Description," *Proc. of the Int. Joint Conf. on Automated Reasoning, IJCAR2006,* U. Furbach and N. Shankar (eds.), No. 4130; *Lec. Notes Art. Intell.,* Berlin: Springer Verlag, 2006, pp. 292–297.

[76] Sirin, E., et al., "Pellet: A Practical OWL-DL Reasoner," *J. Web Semant.*, Vol. 5, 2007, pp. 51–53.

[77] Harris, M., et al., "OBO-Edit: Reasoner Overview," http://wiki.geneontology.org/index.php/OBO-Edit:_Reasoner_Overview: Gene Ontology Consortium, 2008–present.

[78] Parsia, B., et al., "Pellet," http://www.mindswap.org/2003/pellet/: University of Maryland Institute for Advanced Computer Studies, 2003.

[79] Clark, K., and B. Parsia, "Pellet," http://pellet.owldl.com/: Clark & Parsia, LLC, 2008–present.

[80] Knublauch, H., et al., "The Protégé-OWL Plugin: An Open Development Environment for Semantic Web Applications," *Int. Semantic Web Conf., 2004*, Hiroshima, November 7–11, 2004, Stanford University, 2004, http://protege.-stanford.edu/plugins/owl/publications/ISWC2004–protege–owl.pdf, last accessed May 27, 2009.

[81] Humphreys, B. L., and D. A. Lindberg, "The UMLS Project: Making the Conceptual Connection Between Users and the Information They Need," *Bull. Med. Libr. Assoc.*, Vol. 81, 1993, pp. 170–177.

[82] Smith, M. K., C. Welty, and D. L. McGuinness, "OWL Web Ontology Language Guide," http://www.w3.org/TR/owl-guide/: W3C, 2004.

[83] Patel-Schneider, P. F., P. Hayes, and I. Horrocks (eds.). "OWL Web Ontology Language Semantics and Abstract Syntax," http://www.w3.org/TR/owl-semantics/: W3C, 2004.

[84] Patel-Schneider, P. F., P. Hayes, and I. Horrocks, "OWL Web Ontology Language Semantics and Abstract Syntax, Section 5. RDF-Compatible Models," http://www.w3.org/TR/owl-semantics/rdfs.html: W3C, 2004.

[85] Horrocks, I., et al., "SWRL: A Semantic Web Rule Language Combining OWL and RuleML," http://www.daml.org/2003/11/swrl/rules-all.html: W3C, 2003.

[86] Berners-Lee, T., "Semantic Web Road Map," http://www.w3.org/DesignIssues/Semantic.html: W3C, 1998–present.

[87] Donnelly, M., T. Bittner, and C. Rosse, "A Formal Theory for Spatial Representation and Reasoning in Biomedical Ontologies," *Art. Intell. Med.*, Vol. 36, 2006, pp. 1–27.

[88] Chaudhri, V. K., et al., "OKBC: A Programmatic Foundation for Knowledge Base Interoperability," *Proc. of the 15th National Conf. on Artificial Intelligence (AAAI-98)*, Menlo Park, CA: AAAI Press, 1998. pp. 600–608.

[89] Gennari, J. H., et al., "The Evolution of Protégé: An Environment of Knowledge-Based Systems Development," *Intl. J. Hum. Comp. Stud.*, Vol. 58, 2003, pp. 89–123.

[90] Musen, M., et al., "Protégé," http://protege.stanford.edu: Stanford University, 2006–present.

[91] Rothenfluh, T. E., et al., "Reusable Ontologies, Knowledge-Acquisition Tools, and Performance Systems: PROTÉGÉ-II Solutions to Sisyphus-2," *Int. J. Hum. Comp. Stud.*, Vol. 44, 1996, pp. 303–332.

[92] Friedman-Hill, E., "Jess, the Rule Engine for the Java Platform," http://www.jessrules.com/jess/: Sandia National Laboratories, 2008–present.

[93] Racer Systems, "Semantic Middleware for Industrial Projects Based on RDF/OWL, a W3C Standard," http://www.sts.tu-harburg.de/~r.f.moeller/racer/: Racer Systems, 2007.

[94] Zhang, S., O. Bodenreider, and C. Golbreich, "Experience in Reasoning with the Foundational Model of Anatomy in OWL DL," *Pacific Symposium on Biocomputing, 2006,* Altman, R. B., et al. (eds.), Singapore: World Scientific Publishing Co., 2006, pp. 200–211.

[95] Golbreich, C., S. Zhang, and O. Bodenreider, "The Foundational Model of Anatomy in OWL: Experience and Perspectives," *Web Semant.*, Vol. 4, 2006, pp. 181–195.

[96] Goble, C. A., et al., "Transparent Access to Multiple Bioinformatics Information Sources," *IBM Syst. J.*, Vol. 40, 2001, pp. 532–552.

[97] Sterling, L., and E. Shapiro, *The Art of Prolog*, Cambridge, MA: MIT Press, 1986.

[98] Pereira, F. C. N., and S. M. Shieber, *Prolog and Natural Language Analysis*, Stanford CA: Center for the Study of Language and Information, 1987.

[99] Swedish Institute of Computer Science, "SICS Quintus Prolog Manual," http://www.sics.se/isl/quintus/qp/frame.html: Swedish Institute of Computer Science, 1999–present.

[100] Yang, G., et al., FLORA-*2: An Object-Oriented Knowledge Base Language,* http://flora.sourceforge.net/: State University of New York at Stony Brook, 2008–present.

[101] Warren, D. S., "Welcome to the home page of XSB!," http://xsb.sourceforge.net/: State University of New York at Stony Brook, 2008–present.

[102] Yang, G., et al., "FLORA-2: User's Manual," Technical Report, State University of New York at Stony Brook, http://flora.sourceforge.net/, 2008–present.

[103] Chen, H., et al., "F-OWL: An OWL Inference Engine in Flora-2," http://fowl.sourceforge.net/index.html: University of Maryland, Baltimore County, 2003–present.

[104] Kazic, T., "Putting Semantics into the Semantic Web: How Well Can It Capture Biology?," *Pacific Symposium on Biocomputing, 2006,* Altman, R. B., et al. (eds.), Singapore: World Scientific Publishing Co., 2006, pp. 140–151.

[105] Hirtel, D., et al., "Schema Specification of RuleML 0.91," http://www.ruleml.org/spec: W3C, 2006.

[106] Sintek, M., and S. Decker, "TRIPLE—A Query, Inference and Transformation Language for the Semanic Web," Technical Report, DFKI GmbH Kaiserslautern, 2008–present.

[107] Brickley, D., and R. V. Guha (eds.), "RDF Vocabulary Description Language 1.0: RDF Schema", http://www.w3c.org/TR/rdf-schema: W3C, 2004.

[108] van Harmelen, F., P. F. Patel-Schneider, and I. Horrocks (eds.), "Reference Description of the DAML+OIL Ontology Markup Language (March 2001)," http://www.daml.org/2001/03/reference.html: W3C, 2001.

[109] Borgida, A., and P. F. Patel-Schneider, "A Semantics and Complete Algorithm for Subsumption in the CLASSIC Description Logic," *J. Art. Intell. Res.*, Vol. 1, 1994, pp. 277–308.

[110] Plant Ontology Consortium, "Plant Ontology Structure and Definitions," http://brebiou.cshl.org/viewcvs/Poc/ontology/OBO_format/po_anatomy.obo?rev=HEAD&content-type=text/plain: Plant Ontology Consortium, 2008–present.

[111] Wikipedia Volunteers, "Inflorescence," http://en.wikipedia.org/wiki/Main_Page: WikiMedia Foundation, 2001–present, http://en.wikipedia.org/wiki/Inflorescence.

[112] Quine, W. V. O., *Methods of Logic*, Cambridge MA: Harvard University Press, Fourth Ed., 1982.

[113] Ronquist, F., et al., "Morphbank," http://www.morphbank.net: Florida State University, 2008–present.

[114] O'Leary, M. A., and S. G. Kaufman, "MorphoBank 2.7: Web application for morphological phylogenetics and taxonomy," http://www.morphobank.org: State University of New York at Stony Brook, 2008–present.

[115] Rowe, T., et al., "DigiMorph," http://www.digimorph.org: University of Texas, Austin, 2008–present.

[116] Maglia, A., and J. Leopold, "MorphologyNet: Interactive, 3D Visualizations of Animal Anatomy," http://www.morphologynet.org: Missouri University of Science and Technology, 2008–present.

[117] Vega, J. M., et al., "*Agrobacterium*-Mediated Transformation of Maize (*Zea mays*) with Cre-*Lox* Site Specific Recombination Cassettes in BIBAC Vectors," *Plant Mol. Biol.*, Vol. 66, 2008, pp. 587–598.

[118] Chuck, G., et al., "The Maize *tasselseed4* MicroRNA Controls Sex Determination and Meristem Cell Fate by Targeting *Tasselseed6/indeterminate spikelet1*," *Nature Genet.*, Vol. 19, 2007, pp. 1517–1521.

[119] Irish, E. E., "Experimental Analysis of Tassel Development in the Maize Mutant Tassel Seed 6," *Plant Physiol.*, Vol. 114, 1997, pp. 817–825.

[120] Shyu, C.-R., et al., "Searching and Mining Visually Observed Phenotypes of Maize Mutants," *J. Bioinfo. Computnl. Biol.*, Vol. 5, 2007, pp. 1193–1213.

[121] WormBase Consortium, "*C. elegans* Gross Anatomy Ontology," http://www.obofoundry.org/cgi-bin/detail.cgi?id=morm_anatomy: OBO Foundry, 2008–present.

[122] Bard, J., "Mouse Gross Anatomy and Dictionary," http://www.obofoundry.org/cgi-bin/detail.cgi?id=adult_mouse_anatomy: OBO Foundry, 2008–present.

[123] Mabee, P., et al., "Teleost Anatomy and Development," http://www.obofoundry.org/cgi-bin/detail.cgi?id=teleost_anatomy: OBO Foundry, 2008–present.

[124] Leopold, J., et al., "The Amphibian Anatomical Ontology (AmphibAnat)," http://www.amphibanat.org: Missouri University of Science and Technology, 2007–present.

[125] Stein, L., et al., "Plant Ontology Consortium (POC)," http://www.plantontology.org: Plant Ontology Consortium, 2003–present.

[126] Plant Ontology Consortium, "The Plant Ontology Consortium," *Comp. Func. Gen.*, Vol. 3, 2002, pp. 137–142.

[127] Jaiswal, P., et al., "Plant Ontology (PO): A Controlled Vocabulary of Plant Structures and Growth Stages," *Comp. Func. Gen.*, Vol. 6, 2005, pp. 338–397.

[128] Maglia, A. M., et al., "An Anatomical Ontology for Amphibians," *Pacific Symp. on Biocomputing, 2007*, R. B. Altman, et al. (eds.), Singapore: World Scientific Publishing Co., 2007, pp. 367–378.

[129] CARO, "CARO (Common Anatomy Reference Ontology)," http://www.bioontology.org/wiki/index.php/CARO:Main_Page: The National Center for Biomedical Ontology, 2008–present.

[130] UBERON, "UBERON", http://www.bioontology.org/wiki/index.php/UBERON:Main_Page: The National Center for Biomedical Ontologies, 2008–present.

[131] Bodenreider, O., and S. Zhang, "Comparing the Representation of Anatomy in the FMA and SNOMED CT," *AMIA Annual Symp. Proc.,* Bethesda MD: American Medical Informatics Association, 2006, pp. 46–50.

[132] Zhang, S., and O. Bodenreider, "Experience in Aligning Anatomical Ontologies," *Intern. J. Web Semant.*, Vol. 3, 2007, pp. 1–26.

[133] Sneiderman, C. A., T. C. Rindflesch, and C. A. Bean, "Identification of Anatomical Terminology in Medical Text," *Proc. of the AMIA Symp.,* Orlando, FL, November 7–11, 1998, American Medical Informatics Association, 1998, pp. 428–432.

[134] Bean, C. A., T. C. Rindflesch, and C. A. Sneiderman, "Automatic Semantic Interpretation of Anatomic Spatial Relationships in Clinical Text," *Proc. of the AMIA Symp.,* American Medical Informatics Association, 1998, pp. 897–901.

[135] Speretta, M., and S. Gauch, "Using Text Mining to Enrich the Vocabulary of Domain Ontologies," *2008 IEEE/WIC/ACM Int. Conf. on Web Intelligence,* Sydney, Australia, Dec. 9–12, Los Alamitos, CA: IEEE Computer Society Press, 2008.

[136] Speretta, M., and S. Gauch, "Miology: a Web Application for Organizing Personal Domain Ontologies," *Int. Conf. on Information, Process, and Knowledge Management,* Cancun, Mexico, February 1–7, 2009.

[137] Luong, H., S. Gauch, and Q. Wang, "Ontology-Based Focused Crawling," *Int. Conf. on Information, Process, and Knowledge Management,* Cancun, Mexico, February 1–7, 2009.

[138] de Bruijn, J., et al., "Ontology Mediation, Merging, and Aligning," in *Semantic Web Technologies: Trends and Research in Ontology-Based Systems,* J. Davies, R. Studer, and P. Warren (eds.), Chicester, UK: John Wiley and Sons, Inc., 2006, pp. 95–113.

[139] Dou, D., D. McDermott, and P. Qi, "Ontology Translation on the Semantic Web," in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE,* R. Meersman, et al. (eds.), No. 2888, *Lec. Notes Comp. Sci.,* Berlin: Springer Verlag, 2003. pp. 952–969.

[140] Noy, N. F., and M. A. Musen, "PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment," *Proc. of the 17th National Conf. on Artificial Intelligence and 12th Conf. on Innovative Applications of Artificial Intelligence,* Menlo Park, CA: American Association for Artificial Intelligence, 2000, pp. 450–455.

[141] Leopold, J., A. Coalter, and L. Lee, "A Generic, Functionally Comprehensive Approach to Maintaining an Ontology as a Relational Database," *Proc. of the 2009 Intl. Conf. on Ontological and Semantic Engineering (ICOSE 2009)*, Rome, Italy, April 28–April 30, 2009.

[142] Lee, L., et al., "An Ontology Abstract Machine," *Proc. of the 2009 Intl. Conf. on Ontological and Semantic Engineering (ICOSE 2009)*, Rome, Italy, April 28–April 30, 2009.

[143] OBO, "CL: Aligning Species-Specific Anatomy Ontologies with CL," http://bioontology.org/wiki/index.php/CL:Aligning_species-specific_anatomy_ontologies_with_CL: bioontology.org, 2009–present.

[144] Ashburner, M., *Won for All. How the Drosophila Genome Was Sequenced*, Cold Spring Harbor, NY: Cold Spring Harbor Laboratory Press, 2006.

[145] Davidson, D., et al., "EMAP: Edinburgh Mouse Atlas Project," http://genex.hgu.mrc.ac.uk/: University of Edinburgh, 2003–present.

[146] Mouse Genome Informatics Team, "Mouse Genome Informatics: Adult Mouse Anatomical Dictionary Browser," http://www.informatics.jax.org/searches/AMA_form.shtml: Jackson Laboratory, 2008–present.

[147] Lussier, Y., et al., "PhenoGO: Assigning Phenotypic Context to Gene Ontology Annotations with Natural Language Processing," Altman, R. B., et al. (eds.), *Pacific Symp. on Biocomputing, 2006*. Singapore: World Scientific Publishing Co., 2006, pp. 64–75.

[148] Schroeder, M., and Transinsight, "GoPubMed," http://www.gopubmed.com/: Transinsight, 2008–present.

[149] Sahoo, S. S., et al., "An Ontology-Driven Semantic Mash-Up of Gene and Biological Pathway Information: Application to the Domain of Nicotine Dependence," *J. Biomed. Inform.*, Vol. 41, 2008, pp. 752–765.

[150] Smith, B., et al., "Relations in Biomedical Ontologies," *Genome Biol.*, Vol. 6, 2005, p. R46.

[151] National Center for Biomedical Ontologies, "The OBO Relation Ontology," http://www.obofoundry.org/ro/: OBO Foundry, 2006–present.

[152] Kazic, T., "Representation, Reasoning and the Intermediary Metabolism of *Escherichia coli,*" T. N. Mudge, V. Milutinovic, and L. Hunter (eds.), *Proc. of the 26th Annual Hawaii Int. Conf. on System Sciences*, *Vol. 1*, Los Alamitos, CA: IEEE Computer Society Press, 1993, pp. 853–862.

[153] Steeves, T. A., and I. M. Sussex, *Patterns in Plant Development*, Cambridge, U.K.: Cambridge University Press, 1989.

[154] McSteen, P., and S. Hake, "*barren inflorescence2* Regulates Axillary Meristem Development in the Maize Inflorescence," *Development*, Vol. 128, 2001, pp. 2881–2891.

[155] Kiesselbach, T. A., *The Structure and Reproduction of Corn*, Vol. 161, Lincoln, Nebraska: Nebr. Agric. Exp. Stn. Ann. Rep., 1949.

[156] Vollbrecht, E., et al., "Architecture of Floral Branch Systems in Maize and Related Grasses," *Nature*, Vol. 436, 2005, pp. 1119–1126.

[157] Upadyayula, N., et al., "Genetic and QTL Analysis of Maize Tassel and Ear Inflorescence Architecture," *Theo. Appl. Gen.*, Vol. 112, 2006, pp. 592–606.

[158] Gerau, M. and G. Davis. Personal communication, 2008.

[159] Kellogg, E. and L. Stein. Personal communication, 2006.

[160] Ogren, P. V., et al., "The Compositional Structure of Gene Ontology Terms," in R. B. Altman, et al., (eds.), *Pacific Symp. on Biocomputing*, Vol. 9, Singapore: World Scientific Publishing Co., 2004. pp. 214–225

[161] Ogren, P. V., K. B. Cohen, and L. Hunter, "Implications of Compositionality in the Gene Ontology for its Curation and Usage," in R. B. Altman, et al., (eds.), *Pacific Symp. on Biocomputing*, Vol. 10, Singapore: World Scientific Publishing Co., 2005, pp. 174–185.

[162] Wilks, Y., "The Semantic Web: The Apotheosis of Annotation, But What Are Its Semantics?," *IEEE Intell.*, Vol. 23, 2008, pp. 41–49.

[163] Neumann, E. Personal communication, 2006.

[164] Hunter, L. Personal communication, 2006.

[165] Hindmarsh, A. C., "The PVODE and IDA Algorithms," Technical Report, Lawrence Livermore National Laboratory, https://computation.llnl.gov/casc/nsde/pubs/u141558.pdf, 2000.

[166] LAPACK Developers, "LAPACK—Linear Algebra PACKage," http://www.netlib.org/lapack/: University of Tennessee Knoxville, 2008–present.

[167] The MathWorks, *MATLAB Version 6, Release 13*, Natick, MA: The MathWorks, 2002.

[168] Wolfram Research. "Mathematica7," http://www.wolfram.com/products/mathematica/index.html: Wolfram Research, 2008–present.

[169] R Development Core Team, "The R Project for Statistical Computing," http://www.r-project.org/: Wirtschaftsuniversität Wien, 2008–present.

[170] Octave Developers, "Octave," http://www.gnu.org/software/octave/index.html: GNU, 2008–present.

[171] Mabee, P., et al., "Main Page," https://www.nescent.org/phenoscape/Ontologies: nescent.org, 2008–present.

[172] Mabee, P. M., et al., "Phenotype Ontologies: The Bridge Between Genomics and Evolution," *Trends Eco. Evol.*, Vol. 22, 2007, pp. 345–350.

[173] Phenote Developers, "Welcome to Phenote," http://www.phenote.org/: National Center for Biomedical Ontologies, 2008–present.

[174] Mungall, C., "Phenotype Syntax," Technical Report, The National Center for Biomedical Ontology, http://www.fruitfly.org/~cjm/obd/pheno-syntax.pdf, 2006.

[175] Smith, B., J. Köhler, and A. Kumar, "On the Application of Formal Principles to Life Sciences Data: A Case Study in the Gene Ontology," in E. Rahm (ed.), *Data Integration in the Life Sciences, 1st Int. Workshop, DILS 2004, Proc.*, No. 2994 in *Lec. Notes Bioinfo.* Berlin: Springer Verlag, 2004, pp. 79–94.

[176] Schärli, N., et al., "Traits: Composable Units of Behavior," in L. Cardelli (ed.), *ECOOP 2003—Object-Oriented Programming, 17th European Conf., Proc.*, No. 2743 in *Lec. Notes Comp. Sci.*, Berlin: Springer Verlag, 2003, pp. 248–274.

[177] Bairoch, A., "ENZYME," ftp://expasy.hcug e.ch/databases/enzyme, 1992–present.

[178] Rost, B., G. Yachdav, and J. Liu, "The PredictProtein Server," *Nucleic Acids Res.*, Vol. 32, 2003, pp. W321 – W326.

[179] Ofran, Y., et al., "Create and Assess Protein Networks Through Molecular Characteristics of Individual Proteins," *Bioinformatics*, Vol. 22, 2006, pp. e402–e407.

CHAPTER 10

Ontology Applications in Text Mining

Illhoi Yoo and Win Phillips

10.1 Introduction

In this chapter, we discuss primarily how ontologies can benefit text mining. We show the application of ontologies to text mining by providing examples of how ontologies can be used for clustering documents and for mining hidden links from the digital library.

Our examples make use of biomedical ontologies, such as Medical Subject Headings (MeSH) and the Unified Medical Language System (UMLS). This is because they are among the most popular and well known. Among ontologies, biomedical ontologies have been the most studied and developed. The largest digital library in the world and the largest biomedical bibliographic text database, MEDLINE contains more than 19 million articles (as of May 2009). MeSH and UMLS were developed by the National Library of Medicine (NLM) in 1954 and 1986, respectively. In addition, these resources are open to the public and free of charge to use for research purposes; visit the National Center for Biotechnology Information (NCBI), NLM, and National Institutue of Health (NIH) at http://www.ncbi.nlm.nih.gov for details. For further background on this subject, see [15].

10.1.1 What Is Text Mining?

Digital libraries contain a huge amount of text information. For example, MEDLINE, as mentioned in Section 10.1, has nearly 19 million articles. Text-mining techniques have been developed in order to transform this vast amount of text data into machine-understandable information and knowledge. *Text mining* has been defined as the nontrivial discovery process for uncovering novel patterns in unstructured text [2, 6, 8]. Text-mining approaches have been supplemented by techniques and methods from information retrieval (IR), natural-language processing (NLP), data mining, machine learning, and statistics.

10.1.2 Ontologies

Ontologies were discussed in Chapter 1. In this chapter, we make use of the MeSH[1] ontology. Readers who are unfamiliar with MeSH may wish to consult Section 2.2.1 in [15].

10.2 The Importance of Ontology to Text Mining

Traditional text-mining approaches suffer from two serious, fundamental problems, both of which can be resolved by use of an ontology. The first problem is due to the fact that traditional text mining is based on a vector space model,[2] in which each dimension represents a word or a term. In the vector space model, spatial dimensions are independent of one another, so the use of this model as a depiction for text implicitly assumes that in the text document all the words or terms are, likewise, independent. In a text document, however, not all of the words or terms are completely independent; many of the words are about the same topic, so they are related to one another to least some degree. For example, consider elements in the word set {Vehicle, Car, Motor, Automobile, Auto, Ford}. These words are spelled differently, and so a vector space model would treat them as independent, but in a document they might be closely related and sometimes even used synonymously. Likewise, a vector space model would treat elements of the word set {Cancer, Tumor, Neoplasm, Malignancy} as different terms, even though all these words have very similar meanings.

The first fundamental problem, then, is that relying on the vector space model for text processing means that relations among semantically related words or terms (e.g., synonyms or hyper/hyponyms) are not captured. The use of an ontology can alleviate this problem by providing synonym sets of main concepts and by providing concept hierarchies. Text-mining systems would know, then, that *cancer*, *tumor*, and *neoplasm* have similar meanings and are a sort of disease (these terms are classified into *Diseases* in the MeSH tree, or the MeSH term hierarchy).

The second problem is that traditional text-mining approaches do not consider the domain context when processing documents. After converting documents into vector representations, traditional text-mining approaches simply rely on machine learning or mathematical methods, without taking advantage of domain knowledge about the processed text. Because of the complexity of human languages, however, domain knowledge about the text is, in fact, required to properly understand or process documents for text mining. For example, consider a living-room scenario in which one person says to another, "Please turn off the receiver." In the room are a home theater receiver and a wireless mouse receiver. We know the referent of the term "receiver" is the home theater receiver because of domain knowledge; we know the mouse receiver has no on/off switch. A traditional text mining approach does not capture this domain knowledge.

1. http://www.ncbi.nlm.nih.gov/sites/entrez?db=mesh
2. This is because, probably, the vector space model has a strong mathematical background and it has been widely used in machine learning and data mining for decades.

Another example occurs in measuring sentence similarity (one of the most frequently-used techniques in text mining). Consider the following two sentences:

Melatonin is a safe, effective medicine, not requiring a doctor's prescription, for insomnia.

The sleeping hormone supplement is recommended for people with difficulty falling asleep.

Using the traditional vector space model, the similarity is measured as 0%, because the two sentences do not have any significant words in common (*is* and *for* are in both sentences, but they are treated as stop words and eliminated from vector dimensions). We appraise these sentences as nearly the same, however, (i.e., the similarity is around 100%) if we rely on the following domain knowledge:

1. Melatonin is a hormone.
2. Insomnia is a disorder characterized by difficulty falling asleep (sleeplessness).
3. Melatonin is sometimes called a sleeping hormone.
4. A supplement is nonprescription medicine.

In the above example, an ontology can help to measure the similarity between the apparently different, but semantically identical sentences by supplying the relevant domain knowledge: concept hierarchy (the first item), concept definition (second and fourth), and synonym sets (third).

In a domain such as biomedicine, the use of ontologies for text mining is crucial because of high terminological variation (i.e., many synonyms for the same concepts) and complex semantic relationships among terms. Using an ontology is the only way to handle such complex semantic relationships among words or terms in the text, because ontologies supply synonym sets for every concept (e.g., entry terms in MeSH) and hierarchically arrange concepts from the most general to the most specific[3]. The simple use of ontologies in text mining, thus, allows us to easily solve the traditional synonym/hypernym/hyponym problems.[4]

There are other ways in which ontologies can improve on traditional text-mining approaches. By tracking concept hierarchies, ontologies show relationships among terms, thus allowing the measurement of semantic similarities between two different documents. By spanning disparate biomedical information between such documents, automatic hypothesis generation is possible. Ontologies enable *knowledge induction*, the extracting of unknown patterns or rules from particular facts or instances in documents, thus linking new discoveries in biomedical literature to existing biomedical knowledge and promoting knowledge management and ontology learning.

3. The terms in ontologies normally appear in more than one place in the hierarchy, so the terms are actually represented in a graph.
4. Information retrieval also has these problems.

10.3 Semantic Document Clustering and Summarization: Ontology Applications in Text Mining

In this section, we provide an example of how a biomedical ontology can be used for biomedical-document clustering and summarization. For our examples, we continue to use MeSH (a biomedical ontology) and MEDLINE articles.

We believe that optimal text mining requires both document clustering and text summarization, because they are complementary. Since a set of documents is usually has multiple topics, text summarization without document clustering will not yield a high-quality summary. On the other hand, document clustering will allow scant understanding of a set of documents, without an explanation of document categorization or a summary for each document cluster. Thus, a coherent approach to text mining requires both document clustering and text summarization.

We introduce document clustering in Section 10.3.1. In Sections 10.3.2, 10.3.3, and 10.3.4, we discuss a novel graphical representation model, a graph clustering for graphical representations, and a text summarization algorithm, respectively. These are important components of a semantic document clustering and summarization system. After introducing these components, we discuss how the whole system works in Section 10.3.5.

10.3.1 Introduction to Document Clustering

Document clustering is an unsupervised learning process that assumes there is no known information about document similarity. In most cases, even the number of topical groups or clusters (called k) is unknown. Without any prior information about the document set, document clustering groups unlabeled documents into meaningful clusters of similar documents. The number of clusters k may be computed using a cluster validity measure, as in Chapter 3. However, in this chapter we do not address this problem.

Document clustering may be formally defined as follows: Given a set of n documents called DS, DS is clustered into a user-defined number of k document clusters DS_1, DS_2, …, DS_k, that is $(\{DS_1, DS_2, ..., DS_k\} = DS_U)$[5] so that the documents in a document cluster are similar to one another, while documents in different clusters are dissimilar. For purposes of measuring similarities between documents, we use the vector space model, a traditional approach[6]. In this model, each document d is represented as a high-dimensional vector of word and term frequencies (as the simplest form[7]), where the dimensionality indicates the vocabulary of DS.[8]

There are a number of possible similarity measurements for documents. The most widely used similarity method is *cosine similarity*, which is based purely on mathematics. The distance between two vectors is measured by the cosine of the

5. The number of documents in each document cluster is normally different, depending on the con text of the given document set.
6. In this section, we introduce a completely different approach to representing documents, using background knowledge in a domain ontology.
7. Many document-clustering approaches have used TF*IDF, where TF is the frequency of the term and IDF is the inversed document frequency of the term.
8. The size of the vocabulary is the number of distinct words/terms in the document set.

angle between the vectors (please recall that documents are represented as vectors). For example, suppose documents *A* and *B* may be represented as vectors with three dimensions (for simplicity of calculation) as *A*: [1 2 0]; *B*: [2 2 1]. Using cosine similarity, the similarity between *A* and *B* is calculated as about 0.89.

$$\text{Cosine Similarity} = \text{Cosine}\theta = \frac{A \cdot B}{|A| \times |B|} = \frac{(1\times 2)+(2\times 2)+(0\times 1)}{\sqrt[2]{1^2+2^2+0^2} \times \sqrt[2]{2^2+2^2+1^2}} = \frac{2+4+0}{\sqrt[2]{5} \times \sqrt[2]{9}} \approx 0.89$$

Returning to our discussion of document clustering, the grouping of documents occurs through an iterative optimization process, based on the chosen cluster-criterion function. The criterion function measures key aspects of intercluster and intracluster similarities. For example, a criterion function could maximize the sum of the average, maximum, and minimum pairwise similarities among the documents in a cluster. The K-means clustering algorithm, which is widely used in text mining, uses a criterion function that maximizes the similarity between the centroid of a cluster and each document in the cluster. Other possible clustering algorithms for this problem are discussed in Chapter 3, Section 3.2.

10.3.2 The Graphical Representation Model

10.3.2.1 What Are the Advantages of the Graphical Representation Model over the Traditional Vector Space Model?

All document-clustering methods must first convert documents into a proper format, because none of the methods can directly process free text. Since we can recognize documents as a set of concepts that have complex internal semantic relationships, we may represent each document as a graph structure, using the MeSH ontology.

There are a number of good reasons for representing documents graphically. First, graphical representation is a very natural way of portraying document content, because it contains information about the semantic relationships among concepts. In contrast, all such information is lost in a vector space representation. Second, graphical representation provides *document representation independence*; that is, the graphical representation of a document does not affect other representations. In contrast, in a vector space representation, the addition of a single document with new terms usually requires changes to every other document representation. The number of changes required grows dramatically as documents (represented as new vectors) increase. Third, graphical representation guarantees better scalability than the vector space model. Because in text processing a document representation is an actual data structure, for better scalability its size should be as small as possible. As the number of documents to be processed increases, a corpus-level graphical representation expands *at most* linearly and may in fact keep its size, with only some changes in edge weights. In contrast, a vector space representation (i.e., document*word matrix) grows *at least* linearly and may even increase by *n**t* (where *n* is the number of documents, and *t* is the number of distinct

terms in documents). Table 10.1 summarizes these differences between the vector space model and the graphical representation model.

10.3.2.2 How to Create a Graphical Representation of a Set of Documents

We can represent the graph as a triple, $G = (V, E, w)$, where V is a set of vertices that represents MeSH descriptors, E is a set of edges that indicates the relationships between vertices, and w is a set of edge weights that is assigned according to the strength of the edge relationships. The relationships are derived from both the MeSH tree (the concept hierarchy of MeSH terms) and the concept dependencies over documents (discussed in Step 3 below).

The procedure of graphical representation follows three steps: (1) concept mapping; (2) construction of individual graphical representations, featuring both mapped concepts and their corresponding higher-level concepts; and (3) integration of individual graphical representations.

Step 1: Concept Mapping Concept mapping is the mapping of terms in each document to MeSH concepts. Initially, each document must be searched for terms to map, but to reduce unnecessary searches, stop words[9] are first removed and selection is limited to 1- to 3-gram words.[10] The result is a list of 1- to 3-gram words that are candidates for matching to MeSH entry terms. Matching then produces a list of MeSH entry terms. The MeSH entry terms are next replaced with MeSH descriptor terms, to be able to map the synonyms or related terms to MeSH descriptors. The result is a list of matched MeSH concepts. Finally, the system filters out MeSH concepts that in MEDLINE articles are too general (e.g., ENGLISH ABSTRACT) or too common (e.g., HUMAN). We assume that, just like stop words, those terms

Table 10.1 Vector Space Model Versus Graphical Representation Model

	Vector Space Model	*Graphical Representation Model*
Contains Semantic Relationship Information in Documents	No	Yes
Preserves Document Representation Independence	No	Yes
Scalable	No	Yes

9. Stop words (or stopwords) are words such as *a* or *the* that frequently occur in text, but do not bear relevant linguistic meaning, so they are ignored by text-mining systems and text search engines, such as Google, PubMed, and so on. The following sites provide stop-word lists used in PubMed and US Patent DB: http://www.ncbi.nlm.nih.gov/books/bv.fcgi?highlight=stopwords&rid=helppubmed.table.pubmedhelp.T43 and http://www.uspto.gov/patft/help/stopword.htm

10. An *n*-gram word/term indicates that the term consists of *n* words. For example, *high blood pressure* is a 3-gram word or term.

do not have useful distinguishing power to cluster documents. Figure 10.1 provides a depiction of how concept mapping works.

Step 2: Construction of Individual Graphical Representations The next step is that of building individual graphical representations, using concept extension. The mapped MeSH concepts (the output of Step 1) are extended by incorporating higher-level (i.e., more general) concepts from the MeSH tree. Concept extension can make the graphical representation self-contained (or richer) in terms of meaning, which may help users of text-mining systems recognize similar topics. For example, a document containing the concept MIGRAINE may be represented by the concepts {HEADACHE DISORDERS, BRAIN DISEASES, CENTRAL NERVOUS SYSTEM DISEASES} through the concept extension of MIGRAINE. These extended concepts help text-mining systems find similar documents discussing, for example, headache disorders or brain diseases.

In a graphical representation, concept extension creates edges. In the graphical representation, when tracking a concept to a parent concept in the MeSH tree, an edge is drawn between the concept and its higher-level concept. For such new edges, weights are assigned based on their extension lengths. As concepts are extended further, the actual weights assigned to each of the edges decrease in value.

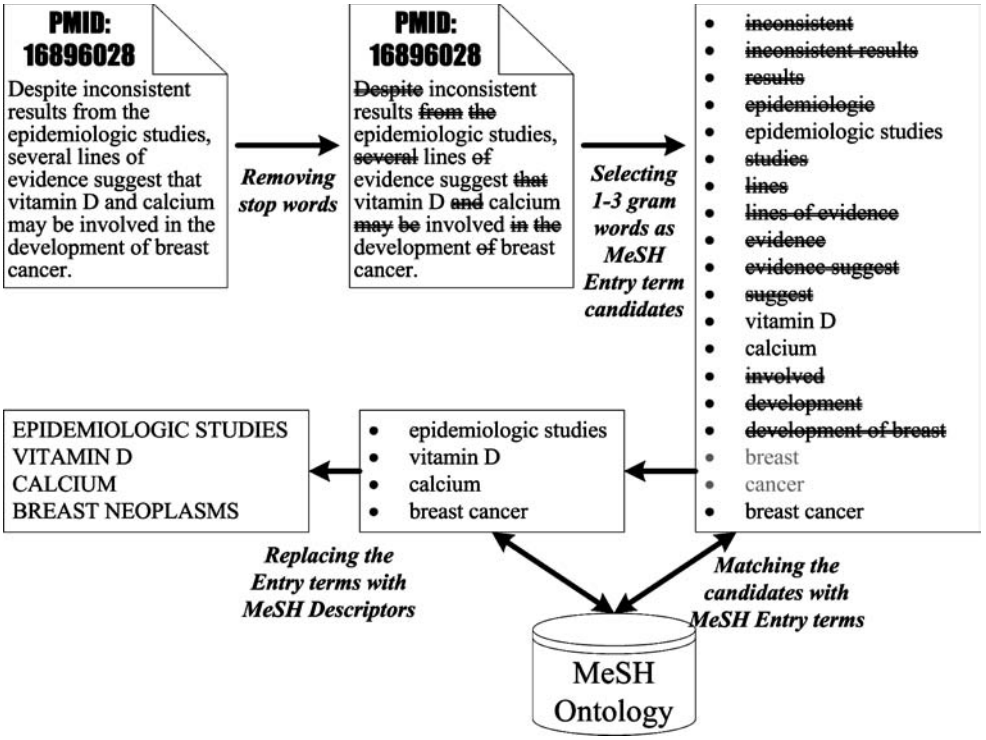


Figure 10.1 An example of concept mapping. The single strikethrough used in "~~of~~" indicates that the word is basically regarded as a stop word, but it could be used as a part of some 3-gram words (e.g., "lines of evidence"). The words "breast" and "cancer" are faded to gray to indicate that each of them is a MeSH term, but because "breast cancer" is also a MeSH term the individual MeSH terms have not been chosen.

This is because increasing generality involves more layers and branches of the concept hierarchy.

We may formalize the weight of the edge generated by a concept extension as follows: suppose α is the parent concept of concept β, and an edge β:α is generated by the concept extension of β. The weight assigned to the edge β:α is based on the absolute locations of concept β and α in the MeSH Tree. This is defined as $\frac{\langle \text{child concept} \rangle}{\langle \text{parent concept} \rangle} = \frac{\langle \beta \rangle}{\langle \alpha \rangle}$ where <concept> is the number of the parent concepts of the child concept, plus the child concept itself. This formula is equal to $\frac{|\alpha \cap \beta|}{|\alpha \cup \beta|}$, which is the similarity of the concepts β and α in the MeSH tree, because $\frac{|\alpha \cap \beta|}{|\alpha \cup \beta|} = \frac{|\beta|}{|\alpha|} = \frac{\langle \beta \rangle}{\langle \alpha \rangle}$. (Recall that because α is the parent concept of concept β, all the parents of α are also β's parents).

Figure 10.2 illustrates this second step of constructing graphical representations using concept extension. Using the MeSH tree, we extend descriptor terms (e.g., {B,C,H} of the document D_1) to their higher-level concepts (e.g., {*A*,*E*,*J*}). Our approach involves higher-level concepts up to the level of (before the) 15 category subroots of the MeSH tree. An example of weighting is presented by edge B:A, which is calculated as $\frac{|\{B,A,E\} \cap \{A,E\}|}{|\{B,A,E\} \cup \{A,E\}|} = \frac{2}{3}$. Some edges, for example A:E (by B and C) and Q:S (by O and Q), are weighted multiple times, their final weight being the sum of all the assigned weights. This multiple weighting emphasizes the relationships between concepts. Note that, in our graphical representation, the thickness of the edge indicates the edge weight: the thicker the line, the heavier the weight.

Step 3: Integration of Individual Graphical Representations Individual graphs generated from each document in Step 2 (i.e., document representations as graphs) can then be merged into a corpus-level graph, which allows co-occurrence concept enrichment. Co-occurrence concept enrichment means semantic relationships are established by concepts co-occurring, when individual document representations are joined in a corpus-level graph. In the corpus-level graph, this creates new edges between the co-occurring variables.

The rationale behind co-occurrence concept enrichment is that the co-occurrence of concepts implies some semantic associations that are not contained in an ontology. We assume that there is a semantic relationship between two concepts if the two concepts are frequently found together in documents, even though an ontology

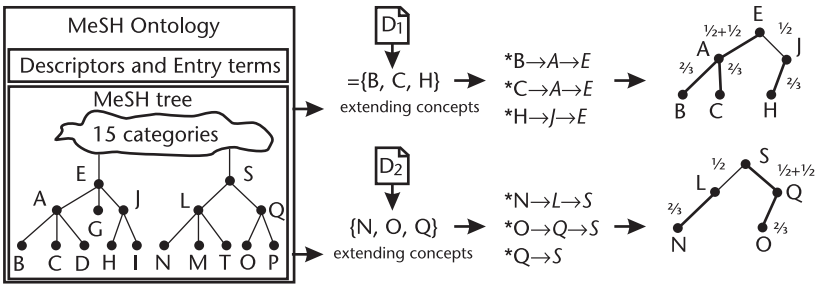


Figure 10.2 Building individual graphical representations.

does not capture this semantic relationship between them. For example, *hypertension* and *diabetes mellitus*, two *type 2* concepts, frequently co-occur in MEDLINE articles (1,850 articles as of 9/14/2008), and there is a semantic relationship between them, because many patients with type 2 diabetes have hypertension.

Recognition of co-occurrence depends on setting a threshold value of term counts at or above which co-occurrence is considered to exist. An issue for co-occurrence concept enrichment is how to set the co-occurrence threshold. We develop a simple algorithm to detect a reasonable threshold value instead of just setting a fixed threshold value. This algorithm finds a bisecting point in one-dimensional data (i.e., a list of co-occurrence term counts) as follows. It sorts the data, takes the two end objects (i.e., the minimum and the maximum values) as centroids, and then successively assigns each remaining object to one of the two centroids, based on the distances between each remaining object and each centroid in a way similar to the *k*-mean clustering algorithm. The centroids are then updated after each assignment by calculating the mean value of the objects for each cluster. The process continues until all objects are assigned to the clusters. The threshold value is then determined as the boundary value between the two clusters.

After obtaining the threshold value, co-occurrence concepts are transformed into edges in the graph, and their co-occurrence counts are used as edge weights. In the graph integration, edge weights total for identical edges. For example, suppose that concepts A and B are co-occurrence terms and their co-occurrence count is 5 (in other words, the concepts are found together in 5 documents). Then, the edge weight between A and B is 5 + (2/3) (see the graph representation of D_1 in Figure 10.2).

Figure 10.3 illustrates this third step of integration of individual graphs. The corpus-level graph is created by merging individual graphs and by transforming co-occurrence concepts into new edges. Note that the integrated graph in Figure 10.3 is based on only four documents (D_1 to D_4) and two co-occurrence concept sets ({C,G}, {T,Q}) from the whole document set (D_1 to D_n). Additionally, Figure 10.3 shows one of the advantages of our approach. Documents D_1 and D_3 do not originally share any common concepts (and thus traditional approaches would not recognize any similarity between those documents), and the same holds for documents D_2 and D_4. But when the documents are represented in integrated graphs,

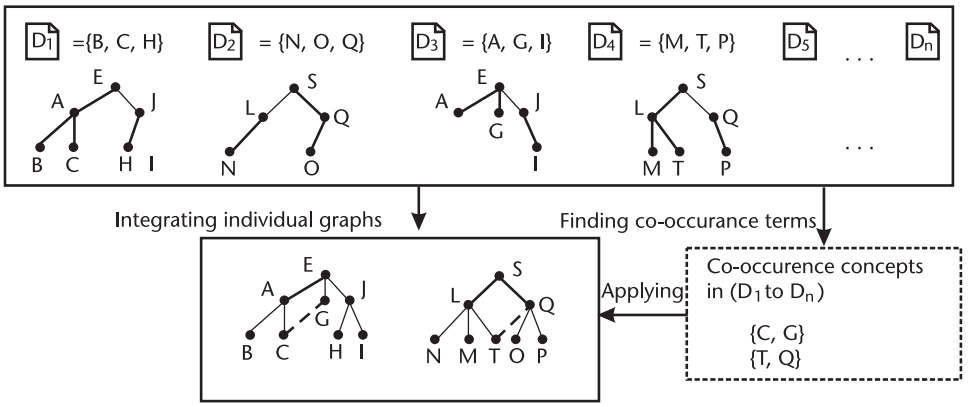


Figure 10.3 Integration of individual graphs featuring co-occurrence concept enrichment.

their graphs can have some common vertices (e.g., {A,E,J} for documents D_1 and D_3, and {L,S,Q} for documents D_2 and D_4). Thus, documents D_1 and D_3 can be regarded as similar to each other, and likewise for documents D_2 and D_4. This newly found similarity occurs because our document representation method involves using higher-level concepts to semantically relate similar documents that do not share common terms.

10.3.3 Graph Clustering for Graphical Representations

After we construct the corpus-level graph for a set of documents, we use graph clustering to find the most relevant higher-level concepts. Ferrer-Cancho and Solé have observed that the corpus-level graph in English follows a scale-free network structure [3]. In other words, only a few words in documents have relationships with the rest of the words, and those words are called *hub* words. Technically speaking, the degree distribution of such a graph decays as a power law, $P(k) \sim k^{-\gamma}$, where $P(k)$ is the probability that a vertex interacts with k other vertices, and γ is the degree exponent [1]. Here, a vertex is a word and a degree is the number of the vertex's relationships with other vertices. In this way, the graphical representation of documents belongs to a highly heterogeneous family of scale-free networks.

The Scale-Free Graph Clustering (SFGC) algorithm that we propose is a clustering algorithm that clusters a graph following scale-free networks. In other words, the algorithm takes advantage of the existence of a few hub vertices (words or terms) in the graphical representation of documents, upon clustering a scale-free network. The SFGC algorithm starts detecting k[11] hub vertex sets (HVSs) as the centroids of k graph clusters and then assigns the remaining vertices to graph clusters, based on the semantic relationships between the remaining objects and k hub vertex sets (the centroids of k graph clusters). Note that the graph-clustering algorithm is conducted on a corpus-level graph, rather than on individual graphs.

Before we describe the SFGC algorithm in detail, we define the following terms:

- *Hub vertices:* In each graph cluster, a set of vertices that are the most heavily-connected in terms of both the degrees of vertices and the weights of the edges connected to vertices (due to the weighted graph). In documents, those vertices are terms that have many strong semantic relationships with other terms.
- *Graph cluster:* A set of vertices that has a stronger relationship with the hub vertices of a graph cluster than with the hub vertices of other graph clusters. A graph cluster is a set of terms that are semantically related to one another.
- *Centroid:* A set of hub vertices, not a single vertex, because we assume a single term as a representative of a document cluster may have its own dispositions, so that the term may not have strong relationships with other key terms of the corresponding cluster. This complies with the scale-free network theory, in which centroids are sets of vertices that have high degrees.

11. k is the number of (graph) clusters and a user-defined value.

The SFGC algorithm proceeds by the following two steps.

Step 1: Detecting k Hub Vertex Sets as Cluster Centroids The main process of the SFGC is to detect k HVSs as the centroids of k graph clusters. As a cluster centroid, an HVS is a set of vertices having high degrees in a scale-free network. In order to measure the centrality degree of vertices in a graph, we use the degree ranking method. This is because degree ranking is fairly comparable to betweenness centrality (BC), and a recent scale-free network study [14] reports that, in finding cluster centroids, betweenness centrality (BC) yields better experimental results than random sampling, degree ranking, and the well-known Hypertext Induced Topic Search (HITS) (introduced by Kleinberg in 1999 [7]). We considered the complexities of BC ($O(|V|^2)$) and the degree-ranking method ($O(|V|)$) in very large graphs (V is a set of vertices in a graph) and selected the degree-ranking method. Unlike other researchers [14], however, who consider only the degrees (i.e., counting edges connected to vertices), we use both vertex degrees and the weights of edges connected to the vertex in the process of ranking vertices. For this purpose, we introduce the salient scores of vertices that are obtained from the sum of the weights of the edges connected to vertices. The salience of a vertex is mathematically rendered as follows:

$$Salience(v_i) = \sum_{e_j \in \{e_j | e_j \text{ having } v_i\}} \text{weight of } e_j$$

where v is a vertex, and e is an edge. After scoring every vertex, the algorithm sorts the vertices in descending order, based on their salient scores, and the top $2*k$ vertices become HVS. Within the top n vertices ($n > 2*k$), SFGC iteratively searches vertices that have strong relationships with any vertices in each HVS. The rationale behind this iterative search process within a limited scope (i.e., the top n vertices) is to gradually expand each HVS, so that majority HVSs do not "eat" minority HVSs (this is called the expansion of HVS in a protected mode). If a vertex has multiple relationships with more than one HVS, the HVS that has the stronger relationship with the vertex is selected. In this way, the top n vertices are assigned to HVSs.

In many cases, HVSs are semantically similar enough to one another to be merged together, because a document set (or a document cluster) may have multiple, but semantically related, topics. In order to measure the similarity between HVSs, we calculate an intraedge weight sum (as a similarity) for each HVS and an interedge weight sum for every possible HVS pair. This mechanism is based on the fact that a good graph cluster should have both maximum intracluster similarity and minimum intercluster similarity. Thus, if an interedge weight sum is equal to or bigger than any intraedge weight sum, the corresponding two HVSs are merged together. This process continues until there are no HVSs that meet the requirement. If the number of HVSs is less than k after the process, SFGC tries to seek a new HVS.

Step 2: Assigning Non-HVS Vertices to Graph Clusters Through this step, all the vertices are grouped, and every HVS becomes a graph cluster. Each of the remaining vertices (i.e., non-HVS) is (re)assigned to the graph cluster to which the vertex is the most similar. The similarity measure used in this step is based on the relationships

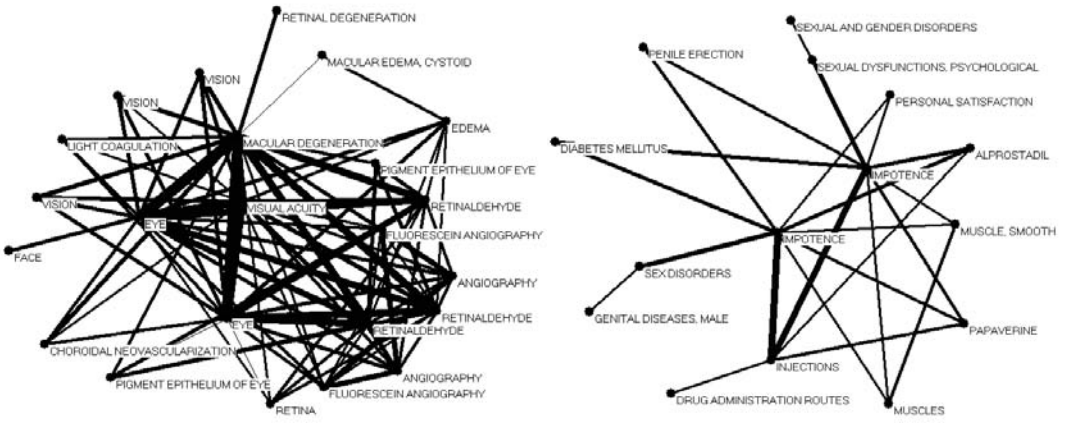
between the vertex and each of the *k* HVSs. The degree of strength of the relationships is measured as the sum of the edge weights. In this way, *k* graph clusters are populated with the remaining vertices.

In order to refine the graph clusters, SFGC iteratively reassigns (non-HVS) vertices to the most similar clusters and updates their centroids (i.e., *k* HVSs), just like K-means updates *k* cluster centroids at each iteration to improve cluster quality. During the updates of HVSs, it uses a bisecting technique, used for the co-occurrence threshold, to select new HVSs from the vertices in each graph cluster (based on their salient scores) by separating the vertices in each graph cluster into two vertex groups (i.e., HVS and non-HVS). Using the new HVSs, the vertices are reallocated to the most similar cluster. These iterations continue until no changes are made to clusters or it stops at a certain iteration.

Finally, SFGC generates both graph clusters and HVSs as cluster centroids or cluster models. Figure 10.4 shows two sample HVSs or cluster models generated from the graph in Figure 10.5. The significant aspects of the use of graphic-document cluster models are that (1) each model captures the core semantic relationship information about document clusters and displays their intrinsic meaning in simple form; and (2) this facilitates the interpretation of each cluster in terms of the key descriptors.

10.3.4 Text Summarization

Text summarization condenses information in a set of documents into concise text. Various scoring mechanisms have been developed to enable selecting and scoring sentences (or phrases), in order to keep some and eliminate others. The key process is how the system selects salient sentences as summary elements. We assume that summary sentences will have strong semantic relationships with other sentences, because summary sentences will cover the main points of a set of documents, and



(a) (b)

Figure 10.4 (a, b) Two sample graphical-document cluster models from the corpus-level graphical representation in Figure 10.5.

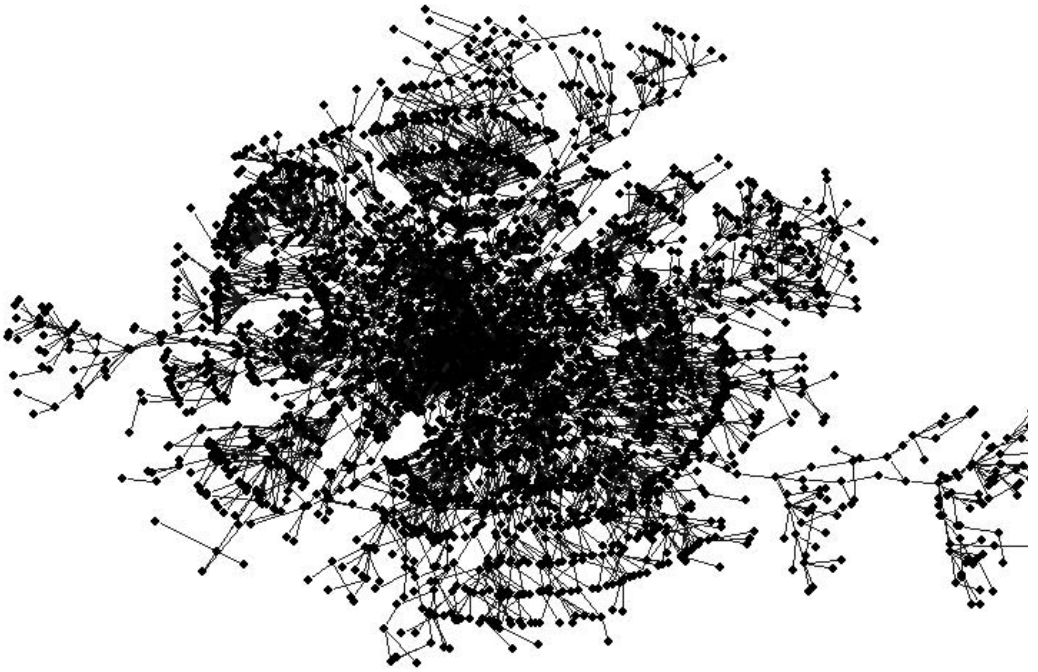


Figure 10.5 A graphical representation of a document set as a scale-free network.

those sentences are supported by other sentences. To this end, in order to represent semantic relationships among sentences, we construct a text semantic interaction network (TSIN), in which vertices are sentences, edges are the semantic relationships between vertices (sentences), and edge weights indicate the degrees of the relationships. For this to work, we need to solve two problems: (1) how to measure similarities between vertices (i.e., sentences); and (2) how, using the similarities, to identify important vertices.

10.3.4.1 How to Measure Similarities Between Vertices in TSIN

To measure the similarities (as edge weights in the network) between vertices (i.e., sentences), we use the notion of *edit distances* between graphical representations of sentences. The edit distance between G_1 and G_2 is defined as the minimum number of structural modifications required to transform G_1 into G_2, where the structural modification is vertex insertion, vertex deletion, or vertex update. For example, in Figure 10.6, the edit distance between the two graphical representations of D_1 and D_3 is 5.

10.3.4.2 How to Identify Important Vertices in TSIN

The next step is how to identify important nodes (i.e., sentences) in TSIN. We basically use a well-known Web-page ranking algorithm, Hypertext Induced Topic Search (HITS) [7], because the problem of identifying important nodes in

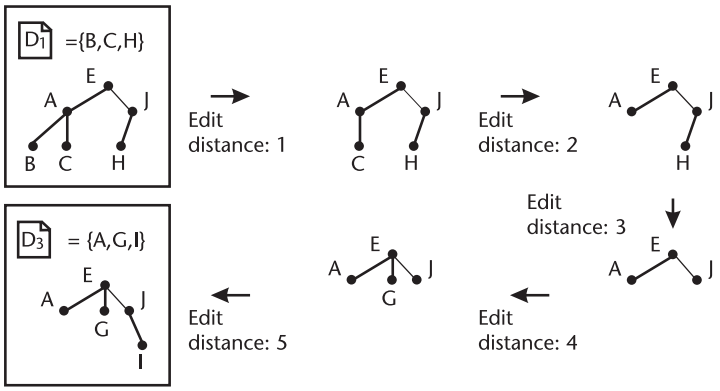


Figure 10.6 Edit distance between two graphical representation of D_1 and D_3.

TSIN is nearly identical to the problem of identifying important Web pages on the Internet.

The HITS algorithm begins with the search constituting the user's query. The search result, consisting of relevant Web pages, is defined as the *root set*. Then, the *Root Set* is expanded to a *base set* by adding two kinds of Web pages: incoming Web pages that have hyperlinks to the *root set* pages and out-coming Web pages that are hyperlinked from the *root set* pages. After the input dataset (i.e., *base set*) is collected, both authority and hub scores are calculated for each Web page in the *base set*. The authority score of a page is based on the hyperlinks *to* the page, while the hub score is based on the links *from* the page. The actual calculations of authority and hub scores are based on the following observations:

- If a page has a high authority score, this means that many pages that have hyperlinks to the page have high hub scores.
- If a page has a high hub score, the page can provide high authority scores to the pages that are hyperlinked by the page.

As indicated, authority scores and hub scores are mutually reinforcing. Based on this intuition, for page i, the authority score $A(p_i)$ and hub score $H(p_i)$ are mathematically rendered as:

$$A(p_i) = \sum_{p_j \in \{p_j | \text{Link}(p_j \to p_i)\}} H(p_j)$$

$$H(p_i) = \sum_{p_j \in \{p_j | \text{Link}(p_i \to p_j)\}} H(p_j)$$

where Link$(p_j \to p_i)$ implies that page $j(p_j)$ has a hyperlink to page i (p_i).

These two iterative operations are performed for each Web page (here, each sentence). The authority score of each Web page is updated with the sum of the hub scores of the Web pages that are linked to the page, and the hub score of each Web page is updated with the sum of the authority scores of the Web pages that link

to the page. The authority and hub scores are then normalized. There are several normalization methods, such as *min-max normalization* or *z-score normalization*. After normalization, values fall within a specified range (e.g., 0–1), so it is easy to see both relative and absolute position or ranking of values.

Unlike hyperlinked Web pages or the Internet, a TSIN graph is an undirected graph, so we may unify authority scores and hub scores into node centrality ($C(N_i)$ for node i), which is mathematically rendered as:

$$C'(N_i) = \frac{C(N_i)}{\sqrt{\sum_i C(N_i)}}, \quad C(N_i) = \sum_{N_j \in \{N_j | Neighbor(N_j, N_i)\}} C(N_j)$$

where, $Neighbor(N_i, N_j)$ indicates that nodes i and j are directly connected to each other and $C'(N_i)$ is the normalized centrality of node N_i, $C(N_i)$. We call this simplified HITS algorithm the *Mutual Refinement (MR) centrality*, since the node centrality is recursively mutually refined. Because the node centralities mutually depend upon one another, we provide each node with its degree centrality as an initial value (otherwise, it would become a chicken-or-egg type of causality dilemma). We apply MR centrality, as well as degree centrality, to measure the centrality of sentences in TSIN. The top n sentences are selected as a summary.

10.3.5 Document Clustering and Summarization with Graphical Representation

This section introduces a novel coherent document-clustering and summarization approach, called clustering and summarization with graphical representation for documents (CSUGAR), and discusses how the whole system works. This approach consists of two main portions, document clustering and text summarization, as shown in Figure 10.7. Three main components of CSUGAR were discussed in Sections 10.3.2, 10.3.3, and 10.3.4. The remaining components are discussed in this section. In Figure 10.7, steps 1–3 correspond to document clustering and steps 4–6 correspond to text summarization.

Step 1: Creating Ontology-Enriched Graphical Representations for Documents and Integrating Them into a Corpus-Level Graph In this step, every document in a MEDLINE document set is represented as a graph through concept mapping and co-occurrence concept enrichment, using a biomedical ontology, MeSH. Individual graphical representations of documents are integrated into a corpus-level graphical representation. Refer to Section 10.3.2 for details.

Step 2: Graph Clustering for Graphical Representation of Documents This graph-clustering algorithm is designed for clustering a scale-free network, such as a graphical representation of documents. The graph clustering algorithm first detects k hub vertex sets as cluster centroids and then recursively assigns nonhub vertices to graph clusters. Refer to Section 10.3.3 for details.

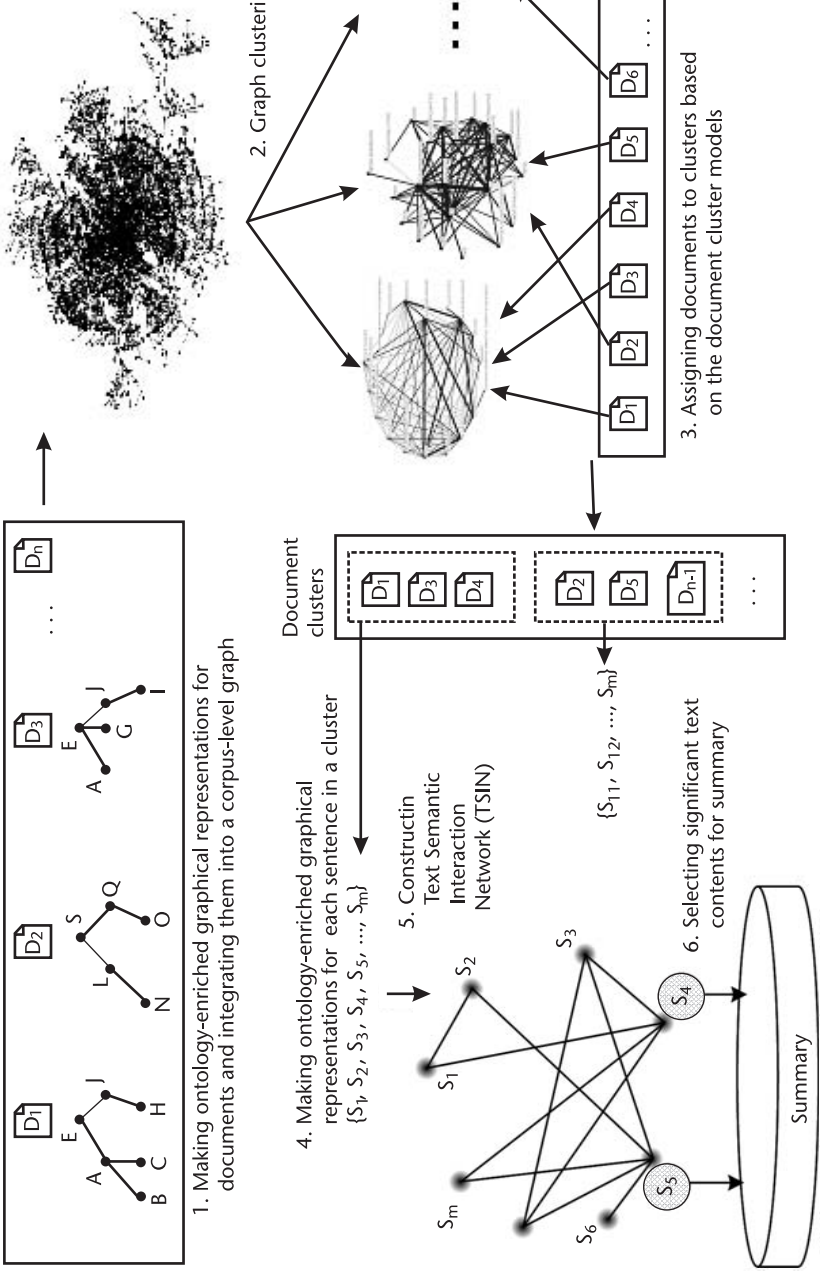


Figure 10.7 The dataflow of CSUGAR.

Step 3: Model-Based Document Assignment In this section, we discuss how to assign each document to document clusters. In order to decide which document belongs to which document cluster, CSUGAR matches the graphical representation of each document with each of the graph clusters as document cluster models or semantics. For this purpose, graph similarity mechanisms, such as edit distance (the minimum number of primitive operations for structural modifications on a graph) could be considered. These mechanisms are not appropriate for this task, however, because individual document representations (graphs) and graph clusters are too different in terms of the number of vertices and edges. As an alternative to graph-similarity mechanisms, we assign documents to graph clusters based on how many of the vertices in the individual graphical representation of each document belong to HVS and non-HVS vertices in each graph cluster. Vertices belonging to HVS are heavily weighted. A document is assigned to a graph cluster based on the highest score among the k clusters.

Step 4: Making Ontology-Enriched Graphical Representations for Each Sentence The graphical representation for sentences is basically the same as the graphical-representation method for documents, except for concept extension and individual graph integration. The concepts in sentences are extended using the relationships among the concepts in relevant document-cluster models, rather than the entire concept hierarchy (MeSH tree), because document-cluster models are richer than the MeSH tree, due to co-occurrence concept enrichment and because document cluster models are regarded as a topic-specific semantic network.

Step 5: Constructing a Text Semantic Interaction Network (TSIN) After representing each sentence as an ontology-enriched graph, a text semantic interaction network (TSIN) is constructed by connecting semantically similar sentences. To measure similarities between sentences, edit distance is used. The purpose of the construction of a TSIN is to identify which vertices (sentences) are important when they are represented in a network in terms of their association; this technique has been widely used in the social-network field. Please refer to Section 10.3.4 for details.

Step 6: Selecting Significant Text Contents for Summary After constructing the TSIN, important vertices (sentences) are identified in the TSIN using the simplified Hypertext Induced Topic Search (HITS) algorithm. Please refer to Section 10.3.4 for details.

10.4 Swanson's Undiscovered Public Knowledge (UDPK)

The huge volume of biomedical literature provides a promising opportunity to increase knowledge by finding novel connections among logically-related medical concepts. For example, Swanson introduced an undiscovered public knowledge (UDPK) model to generate biomedical hypotheses from biomedical literature, such as MEDLINE [11]. According to Swanson, UDPK is "knowledge which can be public, yet undiscovered, if independently created fragments are logically related but never retrieved, brought together, and interpreted." [11]

10.4.1 How Does UDPK Work?

The UDPK model formalizes a procedure to discover novel knowledge from biomedical literature as follows (see Figure 10.8): Consider two separate sets of biomedical literature, *BC* and *AB*, where the document set *BC* discusses biomedical concepts *B* and *C,* and the document set *AB* discusses biomedical concepts *B* and *A*. However, none of the documents in the sets *BC* or *AB* primarily discusses biomedical concepts *C* and *A* together. The goal of the UDPK model is to discover some novel connections between the starting concept *C* (e.g., a disease) and the target concept *A* (e.g., a possible treatment or intervention for the disease) by identifying the biomedical concept *B* (called a bridge concept). For example, Swanson discovered that fish oil (concept *A*) could be a potential treatment for Raynaud's disease (concept *C*) by identifying the bridge concept blood viscosity (concept *B)*. This discovery (UDPK) is accomplished by finding two different biomedical document sets, such that one set (document set *CB*) mentions that Raynaud disease (concept *C*) aggravates blood viscosity (concept *B*), and the other set (document set *BA*) mentions that fish oil (concept *A*) improves blood viscosity (concept *B*).

Swanson's UDPK model can be described as a process to induce "*C* implies *A*", which is derived from both "*C* implies *B*" and "*B* implies *A*"; the derived knowledge or relationship "*C* implies *A*" is not conclusive, but, rather, hypothetical. The concept *B* is the bridge between concepts *C* and *A*. The following steps summarize the procedure [12]:

1. Specify the user's goal (a starting concept *C,* such as a disease, symptom, and so on).
2. Search the relevant documents *BC* from the biomedical literature (e.g., MEDLINE) for *C*.
3. Generate a set of selected biomedical terms (called the *B* list) from the document set *BC,* using a predefined stop-list filter. *B* concepts are chosen from only the titles of the documents.

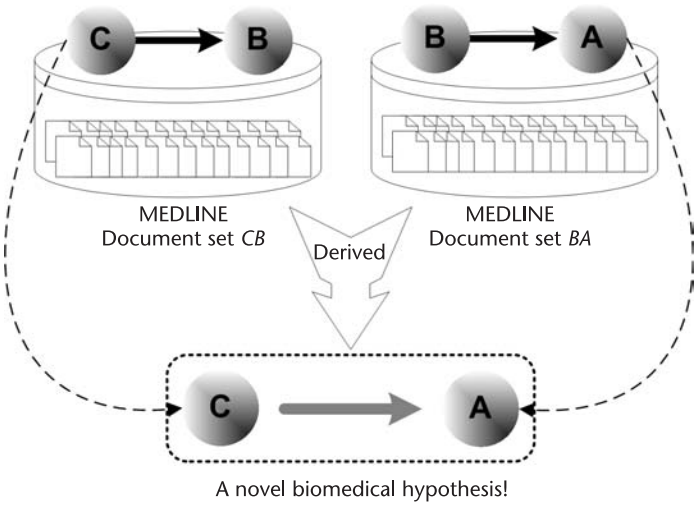


Figure 10.8 Swanson's Undiscovered Public Knowledge Model.

4. Search MEDLINE for each term in the *B* list to retrieve documents *AB*, related to the *B* concepts.
5. Generate a set of biomedical terms (*A* candidates) from the *AB* documents. *A* concepts come from only the titles of the documents.
6. Check whether each of the *A* candidates and concept *C* are cocited together in any MEDLINE articles. If not, keep the *A* candidate.
7. Rank the selected *A* terms, based on how many linkages are made with the *B* terms.

One of the drawbacks of Swanson's method is that it requires a large amount of manual intervention. Although he and his colleague designed an interactive tool called Arrowsmith to automate some of the steps [13], the procedure still requires such manual interventions as having to choose proper lists of stop words and having to filter through a large number of *C*-*B* and *B*-*A* connections to identify the genuinely novel connections/hypotheses. Another problem is that the quantity of relationships or associations among a large number of biomedical concepts is huge, and it grows exponentially as the number of concepts increases. As a result, using an automated process yields too many irrelevant suggestions, and a key issue for the UDPK procedure becomes how to exclude meaningless *C*-*B* and *B*-*A* concept pairs.

Several algorithms have been developed to overcome the limitations of Swanson's approach [4, 5, 9, 10]. However, applied to the UDPK model, none of these approaches considers the different roles of concepts *A* and *B* on concept *C* in filtering terms for concepts *A* and *B*. In addition, the approaches try to tackle the UDPK association problem, using information measures such as TF*IDF, rather than semantic relationships among the concepts.

10.4.2 A Semantic Version of Swanson's UDPK Model

Here, we discuss a semantic-based mining approach called the biomedical semantic-based knowledge discovery system (Bio-SbKDS). Bio-SbKDS automatically mines undiscovered public knowledge from the biomedical literature, using a combination of ontology knowledge and data mining. Specifically, Bio-SbKDS can semantically identify the relationship between concepts *C* (e.g., Raynaud's disease) and *B* (e.g., blood viscosity) and the relationship between concept *B* (e.g., blood viscosity) and *A* (e.g., fish oils) in two sets of biomedical documents and induce a novel hypothesis (i.e., the relationship between concepts *C* and *A*).

There are two key problems in mining the biomedical literature for UDPK: (1) how to determine concept *B* as a bridge concept between concepts *C* and *A*, or, in other words, after retrieving documents related to concept *C* as a starting concept (such as Raynaud's disease), we need to determine which terms (concept *B* candidates) are highly semantically related to concept *C*; and (2) how to determine concept *A* from the many documents retrieved using concept *B*, or, in other words, the document set contains many concepts related to concept *B*, and we need determine which term (concept *A*) is highly semantically related to concept *B* and has a potential yet unpublished relationship with concept *C*. In summary, a major problem of

UDPK is how to properly prune the large number of possible relationships between concepts *C* and *B*, and also between concepts *B* and *A*, in the relevant biomedical literature.

In order to solve these problems, Bio-SbKDS relies on the biomedical ontologies UMLS and MeSH. First, using user-defined semantic relations (possible relationships, e.g., *treats* and *prevents*) between concepts *C* and *A*, Bio-SbKDS induces semantic types as filters for concepts *B* and *A*. Then, using these semantic types (filters), the model finds the correct concepts *B* and *A* from a large number of possible relationships between concepts *C* and *B* and concepts *B* and *A*, respectively, in the relevant biomedical literature. Thus, a major distinguishing feature of this algorithm is that the semantic types for concepts *B* and *A* are automatically derived by using only user-defined semantic relations.

We assume that readers are familiar with basic aspects of UMLS, for example, the notions of concepts, semantic types, and semantic relations. Otherwise, please refer to Chapter 1 or to http://www.nlm.nih.gov/research/umls. Figure 10.9 shows the relations among the UMLS components of concepts (in the metathesaurus), semantic types (i.e., concept categories), and semantic relations (i.e., relationships between semantic types). A concept normally belongs to more than one semantic type, and each semantic type normally has more than one relationship (e.g., *result of*) to other semantic types.

10.4.3 The Bio-SbKDS Algorithm

Figure 10.10 shows the data flow of Bio-SbKDS when mining for UDPK. Each black circled number in Figure 10.10 indicates the procedure step in the algorithm. Next, we explain each step in detail using Swanson's Raynaud's disease example.

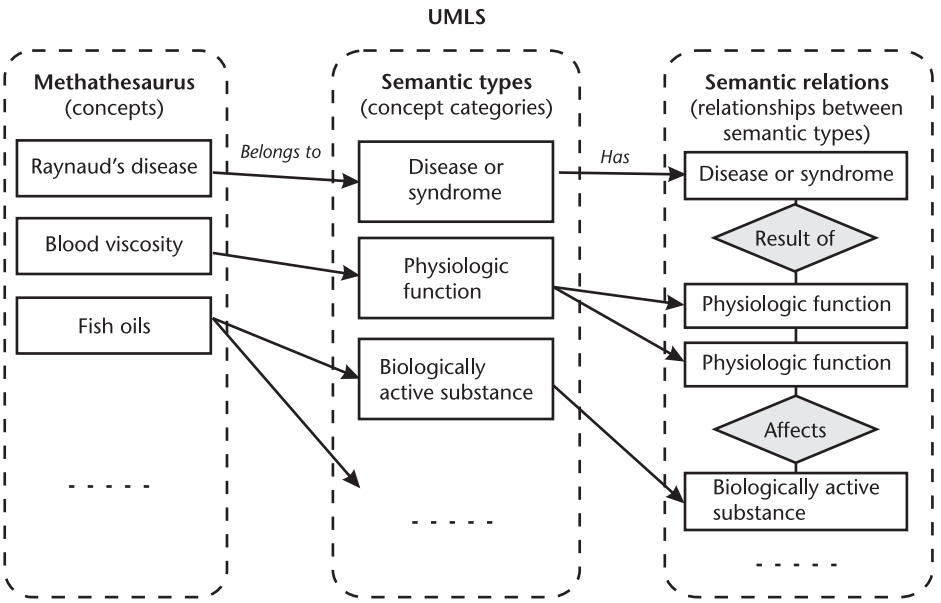


Figure 10.9 The relationship among UMLS components (metathesaurus, semantic types, and semantic relations).

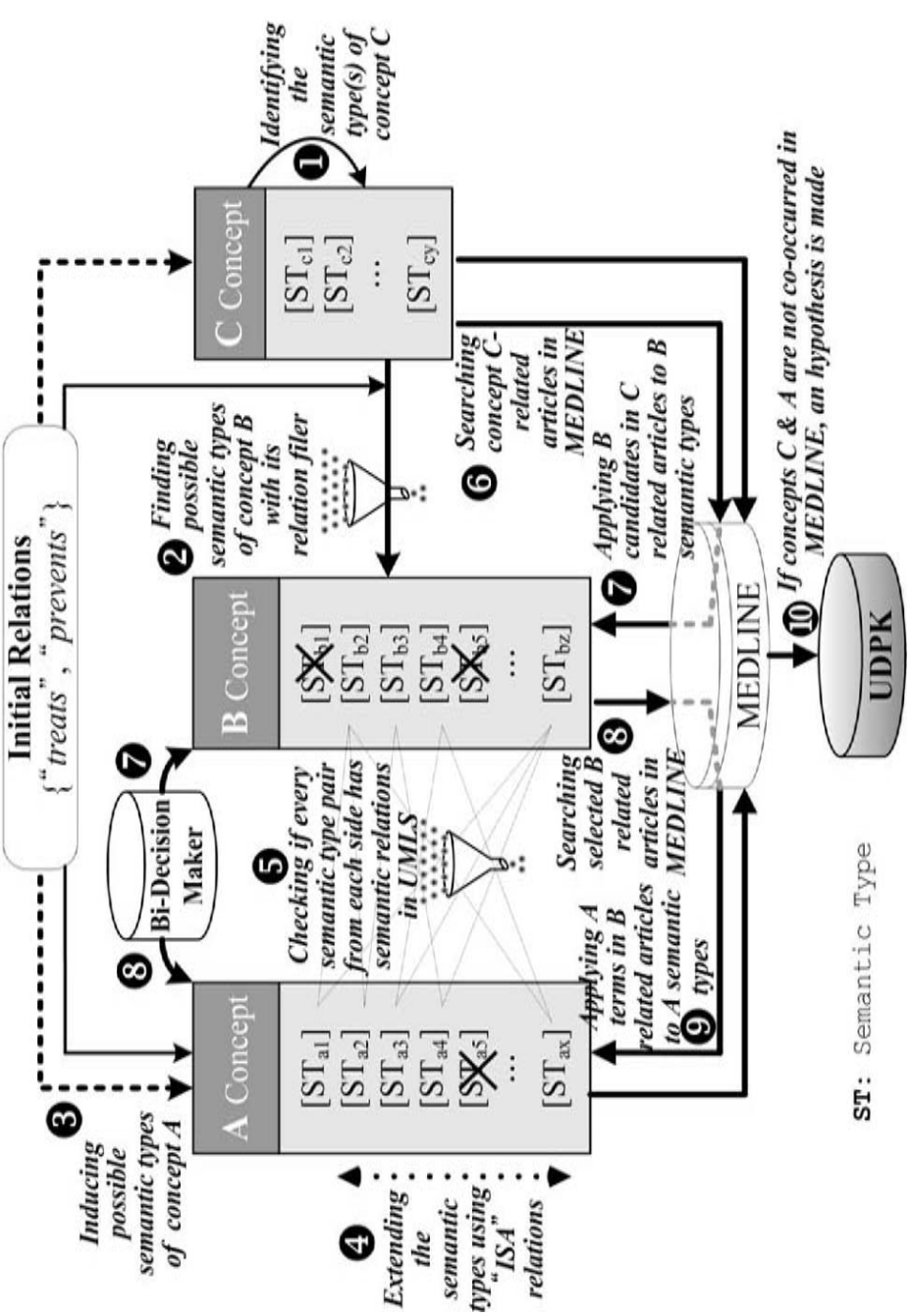


Figure 10.10 The Data Flow of Bio-SbKDS.

Note that in Steps 1–5, Bio-SbKDS first determines what concept *B* and concept *A* should be (i.e., their correct semantic types as concept categories), and then in Steps 6–9, the algorithm identifies the correct concepts *B* and *A*. Finally, to validate the novelty of the hypothesis (i.e., the relationship between concept *C* and concept *A*), Bio-SbKDS checks whether concept *C* and concept *A* have co-occurred in MEDLINE.

The inputs of Bio-SbKDS are the concept *C* (e.g., *Raynaud's Disease*) as a MEDLINE search keyword in MajorTopic MeSH terms, the possible semantic relations between concept *C* and concept *A* (e.g., *treats* and *prevents*), the two predefined relation filters (mainly discussed in Step 2 below), and the role[12] (subject or object) of concept *C* in the semantic relations. Information about the role of concept C is important because knowing the role significantly reduces the search space of semantic relations, and it helps find correct semantic types when inducing semantic types for concepts *B* and *A* from the semantic relations.

Step 1 The semantic type(s) of the starting concept *C* (*ST_C*) is (are) identified through the UMLS semantic network (a UMLS concept may belong to more than one semantic type). Concept *C* must be a MeSH term, because the semantic type of the starting concept is used to induce the semantic types for concept *B*. The semantic type of concept *C* (here, *Raynaud's disease*) is [*Disease or Syndrome*]. The output of Step 1 is a list of the semantic type(s) of concept *C*.

Step 2 In this step, possible semantic types for concept *B* are induced by (1) identifying the initial relations between concepts *C* and *A* and the role of concept *C* in those relations, and (2) applying the predefined relation filter between the semantic type(s) for concept *C* and the semantic type(s) for concept *B*.

How does this step proceed? Before discussing this in detail, we need to mention the internal format of a semantic relation, because we use the format to create a query. A semantic relation has the following format:

<semantic type 1> <semantic relation> <semantic type 2>

Now, we provide several details of Step 2:

1. Identifying the initial relations between concepts C and A and the role of concept C in these relations.

   In Step 1, more than one semantic type for concept *C* can be identified, but not all semantic types identified are going to be valid, because the semantic types of concept *C* must have one of the user-specified initial relations with the semantic types of concept *A*. For this reason, the semantic types of concept *C* must meet the following requirement:

12. For example, if concept *C* is *Raynaud's Disease,* and the semantic relations between concepts *C* and *A* are *treats* and *prevents*, the role of concept *C* (i.e. *Raynaud's Disease*) in the semantic relations is as an *object*, because Bio-SbKDS will try to find something (as concept *A*) that treats or prevents *Raynaud's Disease* (concept *C*).

$$<ST_a> <\{treats, prevents\}> <ST_c>$$

In this way, invalid semantic types of concept C are pruned. Please note that the initial relations are also used in Step 3.

2. Applying the predefined relation filter between the semantic type(s) for concept C and the semantic type(s) for concept B.

   The semantic relation filter between concepts *C* and *B* is {*process_of*, *result_of*, *manifestation_of*, *causes*}. Using this filter, we can create the following queries:

<the semantic types of concept *B*> <*process_of* > <*Disease or Syndrome*>
<the semantic types of concept *B*> <*result_of* > <*Disease or Syndrome*>
<the semantic types of concept *B*> <*manifestation_of* > <*Disease or Syndrome*>
<the semantic types of concept *B*> <*causes*> <*Disease or Syndrome*>

Using these queries (i.e., searching by them in the UMLS Semantic Network), Bio-SbKDS can induce the semantic types of concept *B*, because the semantic types of concept *B* must have at least one of the relations in the relation filters with the semantic types of concept C. Table 10.2 shows that the semantic types *Physiologic Function* and *Steroid* are selected, because the role of concept *C* is set as an object and the relation filter includes *process_of*, *result_of*, and *causes*.

Step 3 In order to derive the semantic types of concept *A*, the initial semantic relations (i.e., *treats*, *prevents*) are used. Here, it is important that concept *C* be set as a subject or an object for the initial relations. For example, if concept *C* is set as an object, only the semantic types on the first (not third) column in Table 10.3 will be considered for the search space for possible semantic types of concept *A*. Note that if a semantic type is too general, then that semantic type will be ignored. Whether or not a semantic type is too general is determined by its hierarchy level in the UMLS Semantic Network.

Currently, levels 1, 2, 3 (e.g., A1.4.1) in the UMLS Semantic Network are regarded as too general (or too broad), since the concepts in the semantic types in such levels are too broad.

Table 10.2 Possible Semantic Types of Concept B That Have at Least One of the User-Defined Semantic Relations (Filters) with the Semantic Type of Concept C

Semantic Types (as Subjects) of Concept B	*Relation*	*Semantic Types (as Objects) of Concept C*
Physiologic function	Process_of	Disease or Syndrome
Physiologic function	Result_of	Disease or Syndrome
Steroid	Causes	Disease or Syndrome

Table 10.3 Possible Semantic Types of Concept A That Have at Least One of the User-Defined Initial Relations with the Semantic Type of Concept C

Semantic Types (as Subjects) of Concept A	*Relation*	*Semantic Types (as Objects) of Concept C*
Antibiotic	Treats	Disease or Syndrome
Drug Delivery Device	Treats	Disease or Syndrome
Medical Device (too General)	Treats	Disease or Syndrome
Pharmacologic Substance	Treats	Disease or Syndrome
Therapeutic or Preventive Procedure	Treats	Disease or Syndrome

Step 4 This step extends the semantic types (of concept *A*), identified in Step 3, through the *ISA* relations.[13] Through this process, all parent (general or broad) or child (specific or narrow) semantic types of the original semantic types of concept *A* are added; however, concepts that are too general are eliminated. For example, the semantic type *Antibiotic* is a child (specific or narrow) semantic type of the semantic type *Pharmacologic Substance*, so *Antibiotic* is added (even though *Antibiotic* is one of the original semantic types). Please note that extended semantic types are used for the semantic types of concept *A* as a category restriction in Step 9.

Step 5 Up to this point, we have induced the semantic types of concept *B* and concept *A*. Are all these induced semantic types valid? One way to prune invalid semantic types uses the fact that concept *B* must have a relationship with concept *A*. In other words, the semantic types of concept *B* must have some semantic relation to the semantic types of concept *A*. If not, the semantic type of *B* is invalid. For example, the semantic type *Organic Chemical* of concept *B* has no relationship with the semantic type *Drug Delivery Device* of concept *A*. Such invalid semantic-type pairs are shown in Table 10.4. In order to detect such unrelated semantic-type pairs, for each semantic type of concept *B*, Bio-SbKDS checks whether there exists at least one relationship between it and any of the semantic types of concept *A*.

If a semantic type of concept *B* does not have a relationship with any of the semantic types of concept *A*, that semantic type is dropped from the semantic-type list of concept *B*. After this process is completed for the semantic types of concept

Table 10.4 Semantic-Type Pairs Having No Relation

Semantic Types of Concept B	*Relation*	*Semantic Types of Concept A*
Invertebrate	None	*Neuroreactive Substance* or *Biogenic Amine*
Geographic area	None	*Neuroreactive Substance* or *Biogenic Amine*
Organic chemical	None	Drug Delivery Device

13. An *ISA* (*is a*) relation in a concept hierarchy indicates a parent (general or broad)-child (specific or narrow) concept relationship between two concepts.

B, the same process is performed for the semantic types of concept *A*. These processes are called mutual qualification.

There is another way to prune invalid semantic types using the predefined relation filter between the semantic type(s) for concept *A* and the semantic type(s) for concept *B*, just as a similar filter was used in Step 2. Bio-SbKDS checks if the two semantic-type sets (for concepts *A* and *B*) pass the predefined relation filter during the mutual-qualification procedure.

This filter includes the following semantic relations: *interacts_with*, *produces*, and *complicates*. Table 10.5 shows the two semantic type sets for concept *B* and concept *A*. Those semantic types are automatically generated using only concept *C*, the initial relations, and the two relation filters.

Step 6 So far, in Steps 1–5, we have discussed what concept *B* and concept *A* should be, in terms of semantic types. Now we discuss how to identify the correct concept *B* and concept *A*, using the semantic types (i.e., concept category filters) from two MEDLINE document sets: set *CB*, which is related to (or retrieved using) concept *C*, and set *BA*, which is related to (or retrieved using) concept *B* (see Figure 10.8). In order to identify the correct concept *B*, Bio-SbKDS searches documents related to concept *C* (in the MajorTopic MeSH terms) in MEDLINE. Bio-SbKDS extracts MajorTopic MeSH terms from the retrieved MEDLINE documents,[14]

Table 10.5 The Semantic Type Sets for Concept B and Concept A

Semantic Types of Concept A (Used as a Category-Restriction Filter)	*Semantic Types of Concept B (Used as a Category-Restriction Filter)*
	Cell Function
	Carbohydrate
	Eicosanoid
	Steroid
	Mental or Behavioral Dysfunction
	Element, Ion, or Isotope
	Organophosphorus Compound
	Congenital Abnormality
	Amino Acid, Peptide, or Protein
	Organism Function
	Pathologic Function
Indicator, Reagent, or Diagnostic Aid	Organ or Tissue Function
Antibiotic	Chemical Viewed Structurally
Biologically Active Substance	Nucleic Acid, Nucleoside, or Nucleotide
Pharmacologic Substance	Organic Chemical
Chemical Viewed Functionally	Cell or Molecular Dysfunction
Immunologic Factor	Inorganic Chemical
Receptor	Acquired Abnormality
Biomedical or Dental Material	Molecular Function
Therapeutic or Preventive Procedure	Neoplastic Process
Vitamin	Mental Process
Hormone	Genetic Function
Enzyme	Lipid
Hazardous or Poisonous Substance	Experimental Model of Disease
Neuroreactive Substance or Biogenic Amine	Physiologic Function

14. This retrieved document set is the MEDLINE document set *CB* in Figure 10.8.

because extracted MeSH terms have strong semantic relationships to concept *C*. We call the extracted MeSH terms candidates for concept *C*. Bio-SbKDS calculates the number of MEDLINE documents containing an extracted MajorTopic MeSH term (a candidate for concept *B*). The count indicates how strongly each candidate is associated with concept *C*.

Step 7 Bio-SbKDS applies the category-restriction filters (i.e., the semantic types of concept *B* obtained in Step 5) to the candidates for concept *B*. If the semantic type of a candidate does not belong to the filter, the candidate is eliminated. During this process, too general candidates are excluded. In addition to those qualifications, Bi-Decision Maker (discussed in Section 10.4.3.1) determines whether each candidate is appropriate to concept *B*. Then, the top *N* candidates are selected in terms of count (i.e., the number of MEDLINE documents containing a candidate as a MajorTopic MeSH term).

Table 10.6 shows the top five candidates for concept *B* in terms of count. *Blood Viscosity* is ranked first, which is the one Swanson found manually.

Step 8 At this point, we need the MEDLINE document set *BA*, mentioned in Figure 10.8, to select candidates for concept *A*. To retrieve the document set *BA*, Bio-SbKDS searches all of the top five candidates for concept *B* in MEDLINE. Table 10.7 shows one of the MEDLINE search keywords for this retrieval.

In order to simulate Swanson's findings, a date range is used, as used for concept *C*. Documents related to concept *C* should be excluded from the MEDLINE document set *BA*, because Bio-SbKDS seeks novel hypotheses (in other words, if concepts *C*, *B*, and *A* have co-occurred together in a MEDLINE document, the relationship between concepts *C* and *A* is not novel).

Table 10.6 Top Five Bridge Concepts, with Their Counts

Candiates for Concept B in MajorTopic MeSH Terms	*Count*
Blood Viscosity	22
Quinazolines	10
Pyridines	8
Vinyl Chloride	8
Imidazoles	8

Table 10.7 A Sample MEDLINE Search Keyword for Retrieving MEDLINE Document Set *BA*

Blood Viscosity[MAJOR]	*1974[dp]:1985[dp]*	*Not Raynaud's Disease[MeSH]*
Concept B in MajorTopic MeSH term	Data range	Excluding documents related to concept C

Next, Bio-SbKDS extracts MajorTopic MeSH terms from the retrieved MEDLINE documents (the *BA* document set in Figure 10.8). The extracted MeSH terms are candidates for concept *A*. Bio-SbKDS calculates the number of MEDLINE documents containing an extracted MajorTopic MeSH term (a candidate for concept *A*). The count indicates how strongly they are associated with concept *C*.

Step 9 This step is basically the same as Step 7. Bio-SbKDS applies the category-restriction filters (i.e., the semantic types of concept *A* obtained in Step 5) to the candidates for concept *A*. During the process, candidates that are too general are excluded. In addition to those qualifications, Bi-Decision Maker (discussed in Section 10.4.3.1) determines whether each candidate is appropriate to concept *B*. Then, the top *N* candidates are selected in terms of the number of MEDLINE documents containing a candidate as a MajorTopic MeSH term.

Step 10 So far, Bio-SbKDS has semantically induced candidates for concept *A*. The last step left is to qualify the candidates for novel hypotheses. As mentioned previously, Bio-SbKDS seeks novel hypotheses, so Bio-SbKDS checks whether concept *C* and any of the candidates for concept *A* have co-occurred in MEDLINE. If concept *C* and a candidate for concept *A* have co-occurred in any MEDLINE document (in other words, basically the two concepts are discussed in an article), they are not regarded as novel. Otherwise, a novel hypothesis is made.

10.4.3.1 Bi-Decision Maker

The most challenging problem in mining for UDPK is how to reduce the number of potential candidates for concept *B*. Because a single candidate for concept *B* may involve many MEDLINE documents (document set *BA*), and this involves innumerable candidates for concept *A*, it is crucial to reduce the number of candidates for concept *B*. Although the semantic types, derived from the initial relations as category-restriction filters, can constrain candidates for concepts *B* and *A*, not every candidate in those semantic types is always appropriate to concepts *B* and *A*. For example, if concept *C* is *Raynaud's disease*, we expect that candidates for concept *B* are symptoms of the disease, something to cause the symptoms, or something directly causing the disease. Consequently, we expect that candidates for concept *A* should be something to relieve the symptoms or inhibit the factors causing the symptoms. The relationship of concepts *B* and *A* should be complementary to the relationship of concepts *B* and *C*. In other words, if concept *C* is a human disease, concept *A* should be something *positive* to the disease, while concept *B* should be something *negative* to humans (the negative entity aggravates the condition of the disease). Therefore, using these properties of concepts *B* and *A*, we can further prune candidates for concept *B* and *A*.

In order to determine whether a MeSH term is positive or negative, the definitions of MeSH terms are analyzed. Currently, our method detects some keywords that have different weights (–5 to 5); minus weights mean negative and plus weights positive. For example, a candidate for concept *B Nifedipine*, which is actually ranked first before the bi-decision qualification process, is dropped after the process, because some terms in the definition, underlined and italicized in Figure

10.11, are positive terms. *Blood Viscosity* is regarded as negative, however, because *morbidity* and *disorder* are negative terms.

Bi-Decision Maker does not always identify all MeSH terms as negative or positive, using their definitions, because NLM does not provide definitions for around 6% of MeSH terms. Secondly, many MeSH terms are between negative and positive. Bi-Decision Maker does significantly reduce the number of irrelevant candidates for concepts *B* and *A*.

10.5 Conclusion

This chapter presents what text mining is and why ontologies are important in text mining. Text mining is a text information discovery process from source text through information retrieval, natural-language processing, information extraction, information induction, information deduction, and/or text summarization. The use of ontologies is becoming standard in text mining, because ontologies can resolve, or at least significantly alleviate, the problems of the vector space model for text mining, and because ontologies help text-mining approaches understand the contents of documents with concept hierarchy, concept definition, and concept synonym sets.

In addition, we have shown two ontology applications in text mining: semantic document clustering and summarization, and the semantic version of Swanson's UDPK model. The coherent approach for semantic document clustering and summarization first represents documents as an ontology-enriched scale-free graph structure, based on the graphical-representation method, using a biomedical ontology. The key to the coherent approach is to construct document cluster models as semantic chunks capturing the core semantic relationships in the ontology-enriched scale-free graphical representation of documents. These document cluster models are detected by considering the term distribution following the scale-free network theory. The models are used for document clustering to assign documents to the best-fit document cluster model. Text summarization constructs a text semantic interaction network (TSIN), using the semantic relationships in the models. Summarization is made of the significant text contents by considering their centrality in the TSIN.

A semantic-based biomedical-literature mining method for Swanson's UDPK is introduced. For a given starting medical concept (concept *C*), Bio-SbKDS discovers

Nifedipine
A potent *vasodilator* agent with calcium antagonistic action. It is a *useful antianginal* agent that also *lowers blood pressure.*

Blood Viscosity
The internal resistance of the BLOOD to shear forces. The in vitro measure of whole blood viscosity is of limited clinical utility because it bears little relationship to the actual viscosity within the circulation, but an increase in the viscosity of circulating blood can contribute to *morbidity* in patients suffering from *disorders* such as SICKLE CELL ANEMIA and POLYCYTHEMIA.

Figure 10.11 Bi-Decision Maker uses the definitions of MeSH terms to determine whether a MeSH term is *positive* or *negative*.

potentially meaningful novel relations or connections with other concepts that have not been published in the medical literature before. The discovered relations/connections can be useful for domain experts to conduct new experiments, try new treatments, and so on. Compared to other approaches, the most significant novel feature of the method is that Bio-SbKDS does not require strong domain knowledge, and it automatically uncovers novel hypotheses or connections among relevant biomedical concepts, with minimum human intervention.

References

[1] Barabasi, A.L., and R. Albet, "Emergence of Scaling in Random Networks," *Science*, Vol. 286, 1999, pp. 509–5.

[2] Fan, W., et al., "Tapping into the Power of Text Mining," *Communications of ACM*, Vol. 49. No. 9, 2005, pp. 76–82.

[3] Ferrer-Cancho, R., and R. V. Solé, "The Small World of Human Language," *Proceedings of the Royal Society of London*, 2001, Vol. 268, pp. 2261–2266.

[4] Hristovski, D., et al., "Supporting Discovery in Medicine by Association Rule Mining in Medline and UMLS," *Medinfo*, Vol. 10, 2001, pp. 1344–1348.

[5] Hristovski, D., et al., "Improving Literature Based Discovery Support by Genetic Knowledge Integration," *Stud. Health Technol. Inform.,* Vol. 95, 2003, pp. 68–73.

[6] Karanikas, H., and B. Theodoulidis, "Knowledge Discovery in Text and Text Mining Software," *Technical Report, UMIST-CRIM,* Manchester, U.K., 2002.

[7] Kleinberg, J., "Authoritative Sources in a Hyperlinked Environment," *Journal of the ACM*, Vol. 46, 1999, pp. 604–632.

[8] Mooney, R.J., and Nahm, U.Y., "Text Mining with Information Extraction, Multilingualism and Electronic Language Management," *Proc. of the 4th Int. MIDP Colloquium*, Bloemfontein, South Africa, September 22–23, 2003, pp. 141–160.

[9] Pratt, W., and M. Yetisgen-Yildiz, "LitLinker: Capturing Connections Across the Biomedical Literature," *K-CAP'03,* Sanibel Island, FL, Oct. 23–25, 2003, pp. 105–112.

[10] Srinivasan, P., "Text Mining: Generating Hypotheses from MEDLINE," *Journal of the American Society for Information Science*, Vol. 55, No. 4, 2004, pp. 396–413.

[11] Swanson, D. R., "Undiscovered Public Knowledge," *Library Quarterly*, Vol. 56, No. 2, 1986, pp. 103–118.

[12] Swanson, D. R., "Two Medical Literatures That Are Logically but Not Bibliographically Connected," *JASIS*, Vol. 38, No. 4, 1987, pp. 228–233.

[13] Swanson, D. R., and N. R. Smalheiser, "Implicit Text Linkages Between Medline Records: Using Arrowsmith As an Aid to Scientific Discovery," *Library Trends*, Vol. 48, No. 1, 1999, pp. 48–59.

[14] Wu, A., M. Garland, and J. Han, "Mining Scale-Free Networks Using Geodesic Clustering," *Proc. of 10th ACM SIGKDD*, Seattle, Washington, August 22–25, 2004; pp. 436–442.

[15] Yoo, I., and M. Song, "Biomedical Ontologies and Text Mining for Biomedicine and Healthcare: A Survey," *Journal of Computing Science and Engineering*, Vol. 2, No. 2, June 2008, pp. 109–136.

About the Editors

Mihail Popescu is currently an assistant professor in the Health Management and Informatics Department at the University of Missouri–Columbia. He obtained his B.S. in electrical engineering at the Polytechnic Institute of Bucharest in 1987. He received an M.S. in medical physics in 1995, an M.S. in electrical engineering in 1997, and a Ph.D. in computer science in 2003, all from the University of Missouri–Columbia. From 1990–1993, he was an assistant professor of electrical engineering at the Bucharest Polytechnic Institute. He worked as a research assistant from 1993–1997 and as a database programmer from 1997–2000 at the University of Missouri–Columbia. He is a member of the Institute of Electrical and Electronics Engineers and a member of the International Society for Computational Biology.

Dong Xu is a James C. Dowell professor and the chair of the Computer Science Department, with appointments in the Christopher S. Bond Life Sciences Center and the Informatics Institute at the University of Missouri. He obtained his Ph.D. from the University of Illinois, Urbana–Champaign in 1995 and completed two years of postdoctoral work at the U.S. National Cancer Institute. He was a staff scientist at the Oak Ridge National Laboratory until 2003 before joining the University of Missouri. His research includes protein structure prediction, high-throughput biological data analyses, and in silico studies of plants, microbes, and cancers. He has published more than 150 papers. He is a recipient of the 2001 R&D 100 Award and the 2003 Federal Laboratory Consortium's Award of Excellence in Technology Transfer. He is a member of the editorial board for *Current Protein and Peptide Science* and *Applied and Environmental Microbiology.* He is a standing member of the NIH Biodata Management and Analysis Panel.

List of Contributors

Troels Andreasen
Roskilde University
Department of Computer Science
P.O. Box 260, DK-4000 Roskilde,
Denmark

Henrik Bulskov
Roskilde University
Department of Computer Science
P.O. Box 260, DK-4000 Roskilde,
Denmark

Armando Blanco
University of Granada
Dept. Computer Science and
Artificial Intelligence
18071 Granada, Spain

Rachel Brenchley
University of Manchester
School of Computer Science
Manchester, U.K.

Carlos Cano
University of Granada
Dept. Computer Science and
Artificial Intelligence
18071 Granada, Spain

Valerie Cross
Miami University
Computer Science & Systems
Analysis Department
Oxford, OH 45066, U.S.A.

Fernando Garcia
University of Granada
Dept. Computer Science and
Artificial Intelligence
18071 Granada, Spain

Andrew Gibson
University of Amsterdam
Swammerdam Institute for
Life Sciences
Amsterdam, The Netherlands

Trupti Joshi
University of Missouri
Computer Science Department
Columbia, MO 65211-2060,
U.S.A.

Toni Kazic
University of Missouri
Department of Computer Science
201 Engineering Building West,
Columbia, MO 65211, U.S.A.

Jennifer L. Leopold
Missouri University of Science and
Technology
Department of Biological Sciences
105 Schrenk Hall
Rolla, MO 65409, U.S.A.

Ping Li
Monsanto Company
1966 Luenenburg Dr.
St. Peters, MO 63376, U.S.A.

Guan Ning Lin
University of Missouri
Computer Science Department
Columbia, MO 65211-2060, U.S.A.

Jingdong Liu
Monsanto Company
800 North Lindbergh Blvd
St. Louis, MO 63167, U.S.A.

F. Javier Lopez
University of Granada
Dept. Computer Science and
Artificial Intelligence
18071 Granada, Spain

Anne M. Maglia
Missouri University of Science
and Technology
Department of Computer Science
307 Computer Science
Rolla, MO 65409, U.S.A.

Win Phillips
University of Missouri
Health Management and
Informatics
One Hospital Drive, MC213,
C053.00
CS&E 720, Columbia, MO 65212,
U.S.A.

Mihail Popescu
University of Missouri-Columbia
Health Management and
Informatics
One Hospital Drive, MC213,
C053.00
CS&E 715, Columbia, MO 65212,
U.S.A.

Jing Qiu
University of Missouri
Department of Statistics
134 I Middlebush Hall
Columbia, MO 65201, U.S.A.

Andy Ross
University of Missouri
Computer Science Department
Columbia, MO 65201, U.S.A.

Zhao Song
University of Missouri
Computer Science Department
Columbia, MO 65211-2060, USA.

Gyan Prakash Srivastava
University of Missouri
Computer Science Department
Columbia, MO 65211-2060, U.S.A.

Robert Stevens
University of Manchester
School of Computer Science
Manchester, U.K.

Lydia Tabernero
University of Manchester
Faculty of Life Sciences
Manchester, U.K.

Katy Wolstencroft
University of Manchester
School of Computer Science
Manchester, U.K.

Dong Xu
University of Missouri
Computer Science Department
Columbia, MO 65211-2060, U.S.A.

Illhoi Yoo
University of Missouri
Health Management and
Informatics
One Hospital Drive, MC213,
C053.00
CS&E 718, Columbia, MO 65212,
U.S.A.

Chao Zhang
University of Missouri
Computer Science Department
Columbia, MO 65211-2060, U.S.A.

Index