CHAPTER 10

A Posteriori Ontology Engineering for Data-Driven Science

Damian D. G. Gessler, Cliff Joslyn, and Karin Verspoor

CONTENTS

10.1 Problem Overview 216
10.2 Vision for the Future 218
10.2.1 Ontology Engineering: Creation and Maintenance 219
10.2.2 Ontology Annotation 220
10.2.3 Ontology Deployment 221
10.2.4 Vision for the Future: Dynamic, Restful, Data-Driven Ontologies 223
10.3 Relationship to Other Technologies 225
10.3.1 Formal Concept Analysis 225
10.3.2 Web Ontology Language 226
10.4 Current State of the Art 227
10.4.1 Formal Concept Analysis Illustration 228
10.4.2 Hybrid Example Combining Data and an A Priori Ontology 230
10.4.3 Mapping to Classes 233
10.4.4 Using OWL 233
10.4.5 World Is Not First Order 238
10.4.6 Making Ontologies Web Savvy 238

10.5 Conclusions and the Path Forward 239
Acknowledgments 240
Abbreviations 240
References 241

10.1 PROBLEM OVERVIEW

Science—and biology in particular—has a rich tradition in categorical knowledge management. This continues today in the generation and use of formal ontologies. Unfortunately, the link between hard data and ontological content is predominately qualitative, not quantitative. The usual approach is to construct ontologies of qualitative concepts and then annotate the data to the ontologies. This process has seen great value, yet it is laborious and the success of ontologies in managing and organizing the full information content of the data is uncertain. An alternative approach is the converse: use the data itself to quantitatively drive ontology creation. Under this model, one generates ontologies at the time they are needed, allowing them to change as more data influences both their topology and their concept space. We outline a combined approach to achieve this, taking advantage of two technologies, the mathematical approach of formal concept analysis (FCA) and the semantic web technologies of the web ontology language (OWL).

Biology has a rich tradition in classifying knowledge that extends back to Aristotle (384–322 BCE) (Aristotle 350 BCE) and more recently to Linnaeus (Linné 1735), Whittaker (1969), and Woese (Woese et al. 1977; Woese et al. 1990). Although these latter examples are primarily taxonomic, knowledge organization today witnesses efforts across the breadth of biology in the creation of controlled vocabularies, ontologies, and knowledge bases (e.g., see online resources at BioPortal,* Open Biological and Biomedical Ontologies [OBO†], and the National Library of Medicine‡).

Biological ontologies, such as those hosted at BioPortal, tend to be predominantly axiomatically weak, deep subsumption hierarchies. "Axiomatically weak" means a preponderance of axiomatic subsumption statements (i.e., subclass of "is a"), and "deep" means successive static relations creating long, transitive chains of class subsumption. These ontologies are noted

* http://bioportal.bioontology.org.
† http://obo.sourceforge.net.
‡ http://www.nlm.nih.gov.

for their relatively few semantic relationships other than subsumptive relations between concepts, relative to the number of classes (del Vescovo et al. 2011).

For perspective, note that the classification system of the Library of Congress (LC) of 21 top-level classes and a few hundred subclasses pales in comparison with the 33,000+ entries in the three major ontologies of Gene Ontology (GO),* not to mention the over 26,000 descriptors and 177,000 supplemental headings in Medical Subject Headings (MeSH†) and the 1 million biological concepts in the Unified Medical Language System's Metathesaurus (UMLS‡). While biological classes tend to be axiomatic and static (i.e., humans conceived the class, named it, and placed it in the ontology in some relation to other classes), LC's classification system classifies any publication in the realm of human knowledge by dynamically generating a part of every item's identifier (the so-called Cutter number). This is equivalent to dynamically generating a class (the set of all publications—e.g., physical copies of a book—that share the same Cutter number) by applying an algorithm to a representative publication's properties to generate the class name.

There is, though, a similarity between the LC system and modern biological ontology creation: the static component of the LC classification—the 21 classes and their subclasses—was built in the late nineteenth and early twentieth centuries using the same methodology that predominates in biology today, a basic methodology of an iterative loop of observation → induction → model creation → deduction. In this methodology, classifiers (1) observe individuals, conceptualized as instances of more general classes; (2) propose the existence of classes, often striving for a certain notion of orthogonality; (3) propose a model (a metaset of classes and relationships between classes) for the assignment of any assemblage of individuals into classes; and (4) implement the model on new individuals, thereby classifying those individuals according to the rules of class assignment. The process is then repeated in refining the classification. The methodology has been productive in biology, but the process is laborious and prone to artifacts, as we discuss in Section 10.2.1. In this chapter, we discuss two approaches that yield a different, and data driven, approach to ontology creation.

* http://www.geneontology.org
† http://www.nlm.nih.gov/pubs/factsheets/mesh.html
‡ http://www.nlm.nih.gov/research/umls/about_umls.html#Metathesaurus

10.2 VISION FOR THE FUTURE

Our vision for the future is a more integrative, scientific information environment. To achieve this, our research and technological approach has been to enable computers to contribute a greater role upstream in data description, discovery, invocation, and response phases. This helps to better position scientists for downstream tasks, such as the interpretation and analysis of data integration products, an area where the human scientific mind is vastly superior to its machine analogs. But deployment of computers upstream means that computers need to be able to access and *assess* data and services with greater degrees of automation. Achieving this requires a computational semantics (from the Greek *semaino*, which means "to mean," and *semantikos*, which means "significant"). A computable semantics is achieved in two tiers: the first uses a nonvariant structural framework with a rigid formal semantics. We use the technologies of Resource Description Framework (RDF) + Web Ontology Language (OWL) + Simple Semantic Web Architecture and Protocol (SSWAP) (Gessler et al. 2009) to produce a semantic web services framework. The second tier uses domain-specific ontologies. We can think of RDF, OWL, and SSWAP as the semantic framework for a web-based computable language; the content—the concepts that actually capture domain-specific statements and their derived inferences—are captured by the use of domain-specific ontologies.

Current-day implementation of domain-specific ontologies—for example, biological ontologies such as those at BioPortal—presents a unique set of challenges for achieving this vision. They can contribute more to biological integration if we can address the following needs: (1) in ontology creation, to ensure that an ontology truly does cover the concept space, minimally and efficiently, and is less susceptible to the conceptual biases of the creators (even if such biases are unintentional and accepted by the scientific paradigm of the day); (2) in ontology maintenance, to have formal support for collaborative editing and quality control, whereby the repercussions of changes to an ontology could be systematized and tested; (3) in ontology annotation, to allow the definition of ontological classes in a logical formalism to inform and drive the process of annotation, that is, assigning individuals to classes; and (4) in ontology deployment, to better enable ontologies to inform the discovery and engagement of semantic web services, thereby contributing to dynamic, on-demand data discovery and integration.

10.2.1 Ontology Engineering: Creation and Maintenance

Ontology creation and maintenance are activities of ontology engineering (Gómez-Pérez, Fernández-López, and Corcho 2003). Although there is no single formula for ontology engineering (Fernández-López and Gómez-Pérez 2002), thoughtful practices have delineated a general process of an environment study, a feasibility study, specification, conceptualization, formalization, implementation, maintenance, and ontology support activities (formula and terminology from Corcho, Fernández-López, and Gómez-Pérez 2007; Fernández-López et al. 1997; see also Sure, Tempich, and Vrandecic 2006). The literature on ontology engineering is well developed. Yet, while surveys of practitioners in the field show general support for formalized processes, 80% of ontology engineering projects reported by Simperl and Tempich (2006) did not follow a specific methodology. Despite the lack of widely adopted methodologies, this general process of human inspection driving ontology creation is widely practiced. We call this general process, as outlined earlier, *a priori* ontology engineering, in reference to the fact that practices across methodologies share the fundamental assumption that ontology engineering should proceed from the informed, human analysis of a problem and that a goal is a formalization of this human conceptualization.

Arguably the most prominent biological ontology is GO (Ashburner et al. 2000). GO is manually constructed and maintained and appears to follow the pattern uncovered by Simperl and Tempich (2006): a formal process is followed, but it is not a scientific methodology as demonstrated by Alterovitz et al. (2010). This work used information theoretic analyses to report topological and structural inefficiencies in GO. Based on this, Alterovitz et al. (2010) proposed a small, restricted set of new classes and relationships. Although their approach was deemed successful in achieving both an improvement in information theoretic measures and operational changes (they proposed 14 recommendations to the Gene Ontology Consortium, many of which have been adopted), the observed effect on a test case of gene expression results is disturbing: "... these changes significantly affected the functional interpretations of 97.5% ($P < .001$) of the experimental gene signatures and altered the resulting set of GO categories by 14.6% on average (p. 130)." In other words, after they improved the information theoretic properties of the ontology by repositioning classes in the subsumption hierarchy, scientific interpretations on data annotated to the ontology changed in 97.5% of the cases. This is disturbing because if such a heavily used ontology—one subject to extensive human inspection and professional use over many years—could be subject to improved topological changes resulting in sweeping reclassifications of sample data, then we are left to wonder to what degree the ontology is organizing knowledge and to what degree it is simply organizing.

The truth is perhaps at neither extreme, but the results suggest that a more empirical process can yield a more robust and actionable knowledge representation. We also note that although there is a compelling basis for information theoretic approaches in ontology engineering, we must still ask how we are to validate their reclassifications. Are there noninformational theoretic criteria that should also be applied? The answer is yes, since we expect exceptions to information-maximizing and entropy-minimizing routines, for example, because of low sample sizes or systematic ignorance of the true, a priori, information model. It appears that the former was explicitly recognized by the Gene Ontology Consortium, as one of the terms recommended to be repositioned in the ontology was not moved due to the low number of extant gene annotations (Alterovitz et al. 2010). The latter—systematic ignorance of the true, a priori, information model—is always implicitly assumed whenever we apply theoretical models to real-world data.

We can conclude that GO is susceptible to alternative topologies, yielding new scientifically relevant classifications, but we cannot conclude yet that these new classifications are optimized over a universe of constraints relevant to knowledge representation. In a way, then, we have the worst of both worlds: our annotation of data to concepts is vulnerable to the topology of the concepts, but we have no guarantee on the "correctness" of the concept topology or concept universe itself.

10.2.2 Ontology Annotation

In biology, gene annotation is the process of assigning a gene—a delineated segment of DNA for which there is evidence that it is transcribed—to a function. Gene annotation is one of the first steps in understanding what a gene does and what specific genes are associated with various biological processes. Genome annotation meetings are often called *jamborees*. The name reflects the activity: knowledgeable, domain experts collaboratively and manually assign genes to functional categories, weighing a variety of qualitative and quantitative evidence to make assignments. Computational assignment is also well practiced, often based on transfer annotation,

but the signal-to-noise ratio is low and computational assignments are accepted as putative until they are supported by experimental evidence.

Currently, for the model plant *Arabidopsis thaliana* (perhaps the best studied model plant) less than 20% of predicted genes have laboratory-based experimental evidence (Goff, pers. comm.). Yet experimental evidence is a weak bar, because experimental evidence is open world: just because a gene is implicated in process *X*, rarely does that mean it is not possible to be implicated in process *Y*. So in gene annotation, negation-as-failure does not apply. Ontology *realization*—the process of assigning individuals to ontological classes—is an activity of gene annotation. However, axiomatically weak subsumption hierarchies are poorly suited to represent negative assertions. For example, many biological ontologies have an expressivity akin to RDF schema (RDFS) semantics for which there exists no notion of negation, inconsistency, or nontrivial satisfiability (all classes are trivially satisfiable in RDFS because there are no negation semantics [Cuenca Grau 2007]). Thus when there is experimental evidence that a gene does not belong to a class, it is difficult or impossible to represent that information in the ontology and the information is lost. Biological ontology annotation is a labor-intensive, human-dominated activity that tends to capture positive class extension (the set of individuals belonging to a class) based on a human interpretation of the data, but it misses extensions that could be derived by negative assertions or inference-derived inconsistencies.

10.2.3 Ontology Deployment

Biological ontologies are recalcitrant to the standards and best practices of the World Wide Web. This limits their contribution to reuse, as well as to emerging technologies such as semantic web services. The problem is threefold.

First, biological ontologies exist primarily as creations of domain-specific human knowledge. As such, they adhere to no web standards per se—they are technology independent; so, for example, there is nothing inherent about the terms, topology, or naming conventions that makes the ontologies web aware. Compare this, for example, with OWL DL (McGuinness and van Harmelen 2004): although the abstract syntax and semantics of OWL DL is also technology independent—it is driven by the axioms and theorems of first-order description logic—implementation standards are tightly linked to web technologies such as international resource identifiers (IRIs) and RDF.

Second, biological ontology engineering is monolithic and opaque to high-throughput, cross effort integration. In general, terms are not formally defined in relation to other terms in other ontologies: terms are de novo to separate ontologies, even if they are derivatives or informally dependent on preexisting concepts. For example, the GO term "Biological Process" (GO:0008150) is a term in GO and the Cell Cycle Ontology (CCO) term Biological Process (CCO:U0000002) is a distinct term in CCO, despite the fact that CCO is by design developed and built from other ontologies, including GO (Antezana et al. 2009). Regardless of how similar or dissimilar the terms' semantics are, they reside in different ontologies as separate, distinct concepts. This creates broad challenges to ontology alignment, because instead of the ontology authors themselves directly using terms from other ontologies, or introducing terms via formal semantic relations to terms in other ontologies, terms are tied tightly to ontology ownership where alignment then proceeds ex post facto. This is evident at repositories of ontology alignments* where secondary information resources such as these mappings are used to describe formal (and informal) alignments across ontologies. Compare this to the IRI and RDF basis of OWL, where it is natural to leverage across domains at the point of term declaration under a formal semantic (e.g., http://thisWebSite.org/thisOntology/thisTerm rdfs:subClassOf, http://thatWebSite.org/thatOntology/thatTerm).

Third, standard ontology packaging practices are not web savvy. Standard practice is to bundle all terms of an ontology into a single file. OWL version of CCO is over 300 MB of uncompressed RDF/XML. CCO terms cannot be independently dereferenced on the web with persistent uniform resource locators (URLs). Access to ontologies is often via non-RESTful (REST stands for Representational State Transfer) user interfaces (Fielding 2000). Even when access is programmatic and RESTful (e.g., BioPortal REST services† or EBI services‡), the returned content may be idiosyncratic (e.g., a nonformal semantics, such as arbitrary XML or HTML). Terms do not use uniform resource identifiers (URIs) for universal addressing; for example, the GO term "Biological Process" exists as a GO concept independent of any web address (GO:0008150); as an OBO term in its 19.4-MB OBO file§ in the 20.6-MB file on the GO website¶ (and

* e.g., http://obofoundry.org/index.cgi?show=mappings.

† http://rest.bioontology.org.

‡ http://www.ebi.ac.uk/QuickGO.

§ http://obo.cvs.sourceforge.net/viewvc/obo/obo/ontology/genomic-proteomic/gene_ontology_edit.obo.

¶ http://www.geneontology.org/ontology/obo_format_1_2/gene_ontology_ext.obo.

other places); as a RESTful, HTML page;* and so forth. When ontologies do use URIs, they may fail to leverage the underlying web capability of dereferencing, for example, the adult mouse brain (ABA) term (#ENT1)† does not resolve to a representation of the term's semantics but to the human readable home page of the *Allen Reference Atlas*. The term's semantics are defined at a separate place, in the ontology itself.‡ The use of the delimiting hash (#) fragment identifier (e.g., #ENTl), instead of a slash (e.g., /ENTl), guarantees that no server can satisfy per-term requests, because the hash is strictly a client-side secondary resource reference (RFC 3986, Section 3.5§). Thus, any server must return the content associated with the reference prior to the hash (e.g., the entire ontology) even if the user agent requests just a single term.¶ These and other factors conspire to render ontologies as artifacts of human knowledge organization, rather than web technologies.

10.2.4 Vision for the Future: Dynamic, Restful, Data-Driven Ontologies

How can we address the limitations mentioned in Section 10.2.3? They are limitations in both ontology engineering and web deployment. We want ontologies to be more quantitative and "objective," that is, subject to algorithmic quantification on the utility of their conceptualization. And we want them to be more informatically aware, such that they leverage the world's dominant distributed informatic infrastructure, that is, the web, to the benefit of data integration. There are ongoing efforts to make biological data interoperable in the context of the semantic web, for example, Bio2RDF (Belleau et al. 2008) and KaBOB (Bada, Livingston, and Hunter 2011); this is in part to address the limitations of current approaches to ontology representation.

Our approach is to invert traditional ontology engineering into what we call "a posteriori" ontology engineering for data-driven science. In a posteriori ontology engineering, one starts with no preconceived concepts. One canvasses the *measurement* technologies of the science to delineate a list of measureable *properties*. Empirical measurement commences, and properties of individuals are assigned observed *values*. Informally, a *class* is defined as "the set of all individuals which share the same properties and

* http://www.ebi.ac.uk/QuickGO/GTerm?id=GO:0008150.

† http://mouse.brain-map.org/atlas/index.html#ENTl.

‡ http://rest.bioontology.org/bioportal/ontologies/download/40133 (API key needed).

§ http://www.apps.ietf.org/rfc/rfc3986.html#sec-3.5.

¶ http://mouse.brain-map.org/atlas/index.html#ENTl.

property values." For continuous values, shared property values encompass binning at an arbitrary resolution. We call this an *informal class*. The network of individuals (datum items) clustered according to shared property values creates a subsumption network, or more formally a lattice: a subclass is a class of individuals that have all the same shared properties and values of its super class, and may be more. This corresponds with an intuitive notion of a class as a set of individuals sharing common properties and values. As a second step (see Section 10.4.3), the definition of a class is formalized to be "the set of all individuals which share necessary and sufficient conditions." We call this a *formal class*. For example, naively, the class of eukaryotes is the set of all individuals such that their cells have the property *hasNucleus* with the value *true*, or the cellular nuclear envelope has *Number of cellular membranes* with the value 2, and so forth. Animals have the properties of eukaryotes and may be more (e.g., are heterotrophic); thus animals are a subclass of eukaryotes in a subsumption lattice built from observed properties and their values.

It may first appear that creating the list of properties has simply shifted the axiomatic class creation of a priori ontology engineering from classes to properties, but this is not the case. The property selection is driven by the measurement methodology—a reflection of the practice and technology of the day. This decouples preconceived notions of what concepts should exist to an empirical canvassing of what properties are being measured. Existential arguments are precluded because the act of measuring itself instantiates the reality of the measurement property and value. Properties may be grouped into a priori subsumption chains, but this is not necessary. We will see that subsumption relations on the data itself are the *result* of data analysis, rather than the a priori scaffold of knowledge organization. Annotation (ontology realization) is the algorithmic process that creates a subsumption lattice, rather than the product of human assignment. A posteriori ontology engineering uses the data itself to create the ontology; thus the ontology is the product, not the assumption. Such an ontology is both temporal and marginal: it is based on the data used to construct it. New data may change both the topology and the realization of the ontology. But the change occurs under the influence of the new data and thus should be a refinement toward a more encompassing model of nature, rather than an undirected change in organization.

We examine a process to achieve a posteriori ontology engineering. The first step uses FCA to create the subsumption lattice from a collection of data with arbitrary measured properties. This creates unnamed, informal

classes, that is, groupings of individuals according to shared properties and values. The second step is to analyze the resultant informal classes for groupings that are deemed scientifically relevant. This is a manual step reliant on human assessment. From this, property restrictions are formalized in OWL, `owl:Restrictions`, which are used to create necessary and sufficient conditions for formal, named classes. The use of necessary and sufficient conditions yields class definitions, and thus concepts are created only to the degree that data support their existence. The existence of formal classes allows new, unclassified data to be annotated to a formal knowledge representation framework based on their observed property values. Lastly, the ontology is deployed web aware, using separately dereferenceable URIs and OWL + SSWAP.

10.3 RELATIONSHIP TO OTHER TECHNOLOGIES

10.3.1 Formal Concept Analysis

A key technology in the order theoretical approach is FCA (Ganter and Wille 1999). FCA produces mathematical methods to represent the hierarchical relationships and implications present among relational data, which are represented as sets of objects and their properties. Within mathematics, FCA derives from the algebraic theory of binary relations and complete lattices. Within computer science, FCA is increasingly applied in conceptual clustering, data analysis, information retrieval, knowledge discovery, and ontology engineering (Ganter, Stumme, and Wille 2005).

FCA defines a formal *context* as a mathematical structure, $K = (G,M,I)$, where G is a set of *objects* (individuals), M is the set of *attributes* (properties) of those objects, and $I \subseteq G \times M$ is an *incidence relation*. The expression $(g,m) \in I$ means that the object g has the attribute m. A formal context can be visualized by a two-dimensional table called a *cross table*, where the presence of a cross in a cell indicates that the object on that row has the attribute on that column. A formal *concept* is a pair of sets (combination of objects and attributes) such that every object has every attribute and every attribute is present on every object. There are often multiple concepts for a given context. We will provide a detailed example in Section 10.4.1.

FCA as a fundamental technique for both construction and integration of ontologies has been known for some years. Bain (2002, 2003) investigated using FCA and inductive logic programming as a means to both identify and create concepts from a formal context. Later, Akand, Bain, and Temple (2007, 2010) extended the approach to GO such that genes could be annotated to derived classes composed of combinations of classes. Cimiano, Hotho, and Staab (2005) provided an example of how formal contexts can be constructed from the output of text parsers. The resulting concept lattices compared favorably to the reference ontologies constructed manually on the same domains. Joslyn, Paulson, and Verspoor (2008) used a similar approach, analyzing a larger general language corpus and focusing specifically on taking advantage of linguistic relations among nouns and verbs to investigate the hypothesis that semantic generality of terms, as represented by hierarchical relations among them, can be determined from the analysis of their shared linguistic contexts. Formica (2006, 2008) first used human-curated ontologies to influence FCA concept similarity measures and then extended the method to substitute an information content metric on the concepts, removing the need for human-curated expertise entirely.

10.3.2 Web Ontology Language

A second technology important for data-driven, a posteriori ontology engineering is OWL (McGuinness and van Harmelen 2004). OWL is the World Wide Web Consortium (W3C)-recommended technology for distributed computational logics on the web. W3C is the voluntary sanctioning body of the World Wide Web. OWL is built on URIs, the more expressive RDF, and the helper technologies of RDFS and XML schema definition (XSD). With OWL, we address both the construction of necessary and sufficient named classes and the deployment of the ontology onto the web (via OWL's tight linkage to the web technologies of URIs, RDF, RDFS, and XSD).

Unrestricted OWL Full is a higher order description logic. As a description logic, its entities are "things" (individuals), relations between things (called properties or predicates), and sets of things (called classes). Individuals are akin to FCA objects as are properties to FCA attributes. OWL, like RDF, allows the expression of properties of classes, classes of properties, and so forth. OWL posses the following important properties: (1) completeness (all truths can be derived), (2) validity and soundness (no falsehoods are derived), (3) monotonicity (inferred truths cannot be later proved not true), (4) nontrivial consistency (no contradictions), and (5) nontrivial satisfiability (logical possibility of a class containing at least one individual). But as a higher order logic it is known to be undecidable, and thus computational inference algorithms are not guaranteed to finish in finite time with finite resources. Yet a few key restrictions on OWL's use

and expressivity yield it first order (called OWL DL), and as such OWL DL gains decidability. Complexity remains high (finite time could still be a long time in worse-case scenarios), but in practice many if not the majority of biological ontologies can be reasoned over in a few seconds (del Vescovo et al. 2011).

OWL's semantics allows one to construct classes via numerous means: axiomatically (simply assert the class); by identity (equivalence and difference); by set operations (intersection, union, complement, enumeration); by specific value, or universal or existential restrictions on properties (*hasValue*, *for all* ∀, and *there exists* ∃); by cardinality constraints on properties (minimum, maximum, and exact); by property characteristics (reflexivity, symmetry, transitivity, and chaining); and so on. This expressivity allows one to state the necessary and sufficient conditions of classes (ontological concepts) based on individuals' observed properties and yields the resultant ontology amenable to computational analysis. An analysis using OWL DL in biology is overall highly favorable, with noted exceptions where DL's fragment of first-order logic (FOL) cannot capture the full breadth of biological expressivity (Stevens et al. 2007).

10.4 CURRENT STATE OF THE ART

FCA creates a concept lattice based solely on objects and their attributes. Objects—data—are grouped by virtue of their shared attributes (properties), and thus these collections of shared properties are essentially unnamed classes, which grow more general (encompassing more objects) as one ascends the hierarchies. Thus, a concept lattice maps to a subsumption lattice. Similar to annotations in manually constructed ontologies such as GO, the process yields data assigned to, or annotated to, classes. But unlike GO, where a class may exist independent of the scientific data (e.g., posited by humans as a placeholder), in FCA the classes are derived automatically and solely from experimental properties of the data. Furthermore, all classes can be proven to be logically consistent, complete, and "minimal," in the sense that if there is no data driving a formal concept's creation there is no consequent class. Most significantly, while there is not necessarily any explicit hierarchical structure in the formal context K, concept lattices derive hierarchical relations among the attributes implicit in the structure of their objects. The resulting concept lattices represent ontological subsumption hierarchies derived from the data. The conspicuous missing element in FCA is that classes remain unnamed (Bain 2002, 2003).

10.4.1 Formal Concept Analysis Illustration

Consider Table 10.1, a cross table showing a brief excerpt of data given by Sjoblom et al. (2006). In this table, the objects, *G*, on the rows are genes or open reading frames (ORFs)—sequences of DNA that are putative, yet not confirmed, genes. (For ease of presentation, we use the word gene to refer to both genes and ORFs). An 'X' appears in the column under the attribute "Breast" if that gene is associated with breast tumors according to the criteria of Sjoblom et al. (2006; espec. Supplementary Material) and the attribute "Colorectal" if it is associated with colorectal tumors. The remaining attributes relate to the cancer mutation prevalence (CaMP) score. The CaMP score is proportional to "… the probability that the number of mutations actually observed in a gene is higher than that expected to be observed by chance given the background mutation rate" (Sjoblom et al. 2006, p. 270). A score of CaMP < 1 is interpreted as evidence that the gene is not implicated in cancer; CaMP ≥ 1 is a decision point implicating a gene in cancer. Out of the total 13,023 genes examined by Sjoblom et al. (2006), 189 have CaMP > 1 and these comprise a class called "CAN-genes" (candidate cancer genes). The use of CaMP score in Table 10.1 illustrates both how nontrivial scientific measurements (complex summaries) may be used as FCA attributes and how continuous variables may be binned on scientifically relevant thresholds into a discrete cross table.

Identifying relevant patterns such as which collections of genes are highly, somewhat, or not implicated in breast, colorectal, or both forms of cancer involves sorting the table by rows and columns multiple times to move checkboxes together. Some patterns are obvious: all genes are associated with colorectal cancer. Others are more subtle: every gene associated with breast tumors is also associated with colorectal tumors, yet it does not induce disease. This simple example is illustrative, but as the number of objects and attributes increases (the real data set involves

TABLE 10.1 Cross Table Summarizing Five Attributes of Two Genes and an ORF

	Tumor		CaMP		
	Breast	**Colorectal**	**<1**	**≥1**	**≥1.5**
SKIV2L	X	X	X		
C6orf29		X		X	
SLC29A1		X		X	X

Source: Sjoblom T et al., *Science*, 314(5797):268–274, 2006. With permission.

thousands of genes) the complexity of permuting and sorting such tables to find all patterns is combinatorially overwhelming for manual inspection.

The formal context *K* on objects (rows: genes) and their attributes (columns: tumor types and CaMP levels) summarizes all possible permutations of rows and columns to identify *maximal rectangles* of checkboxes in its creation of a concept lattice, a semantic hierarchy that dually catalogs both the collections of objects that have certain attributes and the collections of attributes that hold for certain objects. An example concept lattice is shown in Figure 10.1. In the lattice, nodes contain information about precise facts in Table 10.1, such as the fact that SKIV2L is the only gene that is both implicated in breast tumors and not disease related. Similarly, one can examine attributes such as CaMP ≥ 1 to show that C6orf29 is disease implicated, but traversing downward we see that so is SLC29A1 (so that C6orf29 has only CaMP ≥ 1, and not CaMP ≥ 1.5). Going back the other way, both genes are associated with colorectal tumors and only colorectal tumors. Finally, information is available about what pairs or groups of objects and attributes are in common. For example, to see what SKIV2L and SLC29A1 have in common we traverse upward from both until we arrive at their *join* at "Colorectal" tumor: not only are both genes implicated in colorectal tumors (though only SLC29A1 is additionally

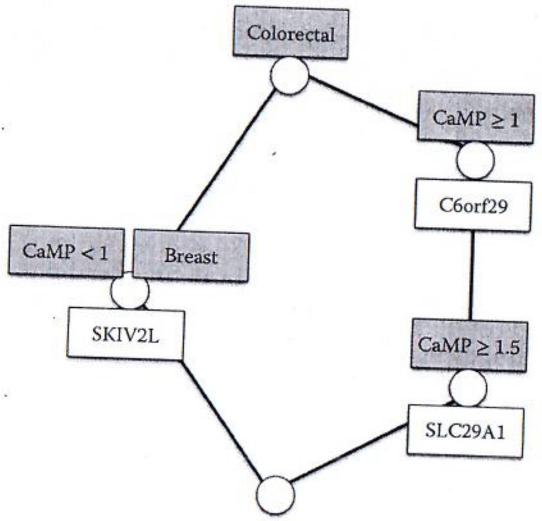

FIGURE 10.1 Concept lattice for Table 10.1.

implicated as disease associated) but also that is *the only thing* they have in common.

The example also shows the marginal effect of limited data sets. Among the genes present in the formal context of Table 10.1 and the concept lattice of Figure 10.1, genes involved in breast tumors "imply" genes involved in colorectal tumors, that is, breast tumor is below (is subsumed by) colorectal tumor. Yet this is not true for the full data set (data not shown). Thus as more data is added, scientific conclusions may also alter, reflecting the new data's contribution. Yet in marked distinction with the situation described by Alterovitz et al. (2010) and GO, subsequent changes to a resultant ontology built from the formal context necessarily reflect the *data's impact* on new knowledge implications, rather than causing a reorganization of a priori concepts.

10.4.2 Hybrid Example Combining Data and an A Priori Ontology

A priori domain ontologies operate by coding the fundamental concepts and semantic relations of a domain and, as such, they may still contribute a scaffold for knowledge representation. FCA can leverage data against this scaffold to further refine the ontologies. Figure 10.2 shows an example of a small portion of GO, which currently holds over 36,000 such categories. Each functional category is adorned with some of the gene products that perform those functions. For example, the gene *Mcmd4* in mice performs (nonexclusively) the function "DNA ligation," and thence by subsumption the functions "DNA-dependent DNA replication," and so forth. Such structures need to be maintained manually, which is both difficult and error prone.

In addition to pure induction of taxonomic structures from underlying relational data, FCA can be used to combine hand-crafted ontological structures with automatically constructed hierarchical information (see the studies of Kaiser, Schmidt, and Joslyn [2008a, b] and Guo et al. [2011] for a more complete formal mathematical consideration). For example, extant GO annotations can be used as attributes on the data. Consider again the GO fragment from Figure 10.2, and now use it to construct a formal context by making the objects (rows) the gene products (e.g., *Mcmd4*) and the columns GO functional categories (e.g., "DNA ligation"). While these relationships alone will determine a hierarchical structure using FCA, the subsumption implications in GO that incorporate its manually constructed taxonomic structure are represented in the columns.

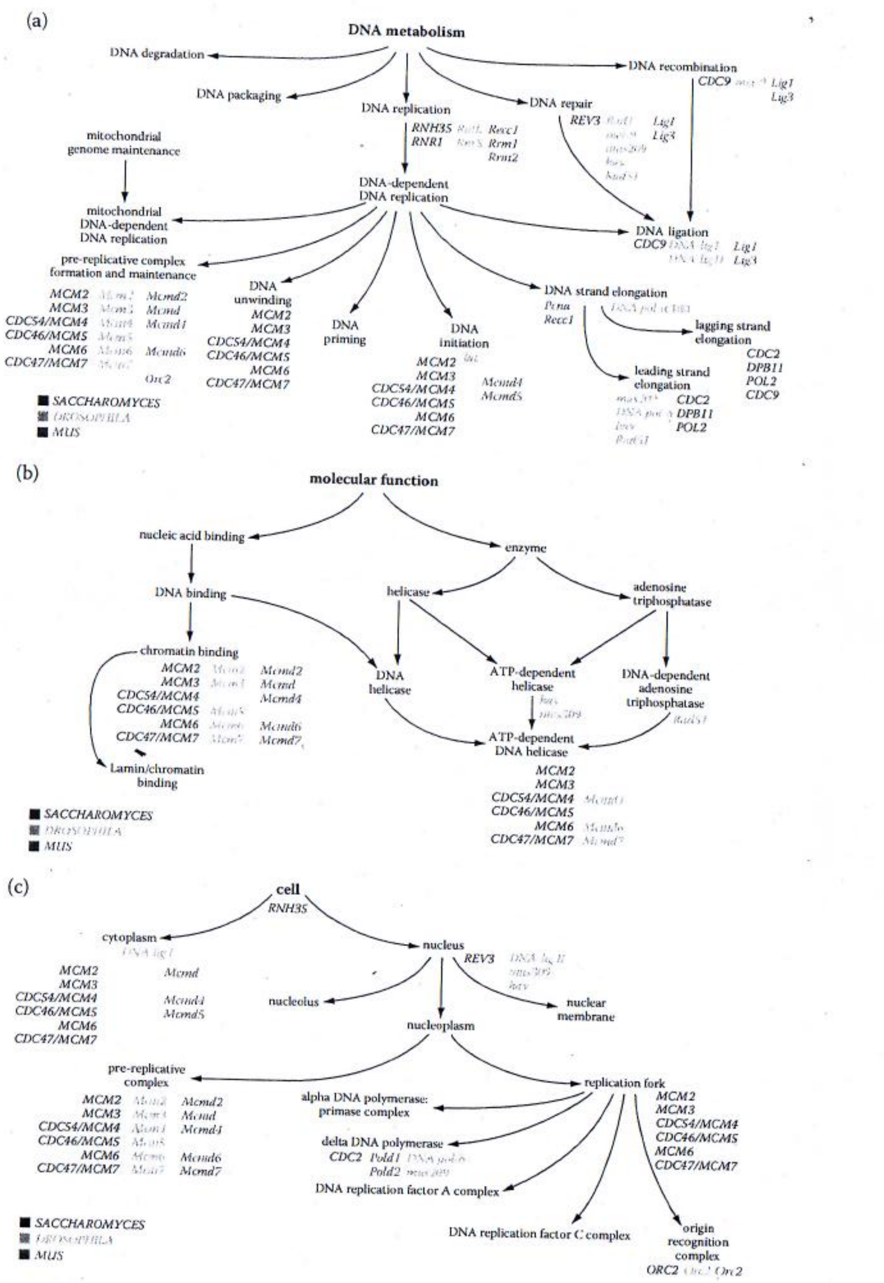

FIGURE 10.2 A portion of Gene Ontology, Biological Process. (From Ashburner, M., et al., *Nature Genet.*, 25(1), 25–29, 2000.).

The result is an adjustment of the original hierarchy, as shown in Figure 10.3. Note, for example, that "DNA initiation" is now both a child of "DNA dep. DNA replication" and a parent of "DNA unwinding," whereas before it was only a child of "DNA dep. DNA replication." This is because of the great amount of annotation overlap between these categories. Similarly, "DNA repair" is now a parent of "DNA recombination," because of the common annotation with the Lig 1 and Lig 3 genes for mouse.

FCA has limitations. Subsumption only occurs to the extent that objects share attributes; thus, it is trivially uninformative if objects use different attribute tags for equivalent attribute concepts. If objects represent ambiguous concepts, they will be associated with sets of attributes that conflate multiple meanings. Thus, FCA benefits from the consistent use of a controlled vocabulary. Most FCA algorithms are optimized for discrete

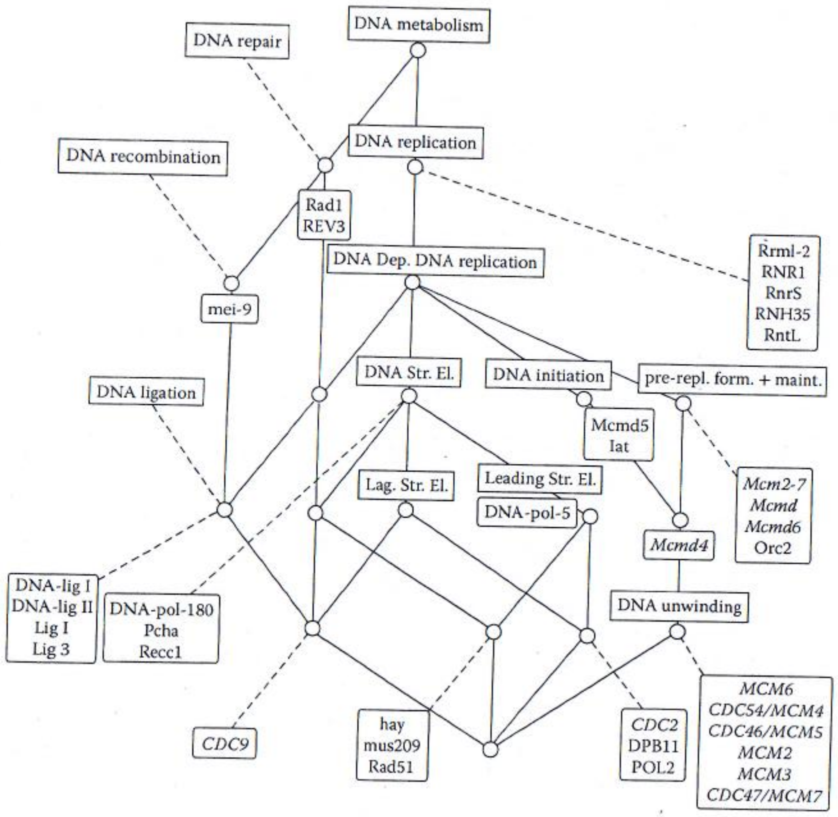

FIGURE 10.3 Functional categories of the Gene Ontology fragment from Figure 10.1 adjusted by their genomic annotations.

characters. Continuous values can be addressed by first binning values into a finite set of value ranges or by ranking. Either transformation loses information. Additionally, as the number of individuals and attributes grows it is computationally expensive to compute the subsumption lattice. Finally, subsumption implies necessary but not sufficient conditions, so FCA alone is not sufficient to generate formal class definitions. Although none of these limitations appear to be fatal to the use of FCA in ontology engineering, they do hint that more research is warranted before FCA can be deployed in a production environment.

10.4.3 Mapping to Classes

From a data-driven subsumption lattice, we now seek to map necessary relations to formal necessary and sufficient class definitions. We proceed by examining each class from the FCA subsumption lattice and its shared properties (attributes). Our goal is to identify those classes that capture concepts that we value sufficiently so as to name and construct necessary and sufficient conditions. This may be done manually, or by identifying topological or informationally rich parts of the concept lattice (Bain 2003; Formica 2008). The process is nondestructive (we do not dismantle the source subsumption lattice); it may be quantitative (e.g., we may use threshold criteria such as the presence/absence of properties, the size of class extension, or the position of the class in the lattice), and it may be qualitative (studying a specific disease drives our relative valuation on which classes to name). The process may be done statically (once, to produce an authoritative model of the data) or dynamically (as affected by specific questions on the data).

10.4.4 Using OWL

Once key concepts are identified, class definition constructs of OWL can be used to create class restrictions. To create necessary but not sufficient conditions, we use subsumption, that is,

```
<owl:Class rdf:about = "&dataOntology;Eukaryota">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource =
              "&propertyOntology;hasNucleus"/>
            <owl:hasValue rdf:datatype =
              "&xsd;boolean">true</owl:hasValue>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

The above W3C-recommended RDF/XML snippet serialization declares the class `Eukaryota` to be a subclass of the anonymous class of all individuals that have the `hasNucleus` property with a value of `true` (see the work of Stevens et al. [2007] for exceptions). Informally, this can be serialized in Notation 3 (N3) without loss of information:

```
dataOntology:Eukaryota
    a owl:Class ;
    rdfs:subClassOf [
        a owl:Restriction ;
        owl:hasValue "true"^^xsd:boolean ;
        owl:onProperty propertyOntology:hasNucleus
    ].
```

To create necessary and sufficient conditions (a formal class definition), we use equivalence:

```
dataOntology:Eukaryota
    a owl:Class ;
    owl:equivalentClass [
        a owl:Restriction ;
        owl:hasValue "true"^^xsd:boolean ;
        owl:onProperty propertyOntology:hasNucleus
    ].
```

Some care needs to be exercised when using equivalence because it is possible that the restrictions will be uncovered by a DL reasoner to be equivalent to `owl:Thing`—the top OWL concept to which all concepts, properties, and individuals belong. Informationally this is redundant, whereas computationally it can be expensive: equivalency is transitive, and thus equivalence to `owl:Thing` can generate excessive statements that unnecessarily burden computational analysis. Equivalence to `owl:Thing` is informationally vacuous and should be removed.

Yet the key motivation for definitions (necessary and sufficient conditions), versus just subsumption (necessary conditions), is that new, unannotated data can be assigned to such classes. Thus, one uses an initial set of experimental data to construct the FCA concept lattice and the resultant

ontology to annotate new data under the logical requirements for class extension. At some point, this new data can also be iterated back to the FCA step to revise the underlying concept lattice.

From just a few named classes, one can define new classes based on logical relations. For example, consider the property `hasNucleatedCell`, which is an object property that connects a subject individual to an object datum—in this case, the representation of a `NucleatedCell`, that is, a cell with a nucleus. Thus, we may define

```
dataOntology:Eukaryota
    a owl:Class ;
    owl:equivalentClass [
        a owl:Restriction ;
        owl:onProperty propertyOntology:has
          NucleatedCell ;
        owl:someValuesFrom dataOntology:NucleatedCell
    ].
```

The existential quantifier *there exists* ∃ is modeled with OWL construct `owl:someValuesFrom`. Thus, any individual with the observed property `hasNucleatedCell` with a value belonging to the class `NucleatedCell` will be assigned to the class `Eukaryota`; that is, it is a eukaryote. We can force the range of `hasNucleatedCell` to always be `NucleatedCell` with the following property definition:

```
dataOntology:NucleatedCell
    a owl:Class.

propertyOntology:hasNucleatedCell
    a owl:ObjectProperty ;
    rdfs:range dataOntology:NucleatedCell.
```

The objects of `hasNucleatedCell` will be inferred to belong to the class `NucleatedCell` if they are not already so declared. Axiomatic assignment to classes can occur on both the range and the domain. Depending on our data collection quality control, if we want to state a priori that anything with the `hasNucleatedCell` property is axiomatically a eukaryote then we can enforce this by adding a global domain:

```
dataOntology:NucleatedCell
    a owl:Class.
```

```
dataOntology:Eukaryota
    a owl:Class.

propertyOntology:hasNucleatedCell
    a owl:ObjectProperty ;
    rdfs:domain dataOntology:Eukaryota ;
    rdfs:range dataOntology:NucleatedCell.
```

Axiomatic domains and ranges should be used judiciously because they force a reasoner to deduce that a datum belongs to the respective class regardless of other statements. Alternatively, we may choose to define eukaryote (and thus its class extension) solely a posteriori, in which case we would not put a domain on `hasNucleatedCell` but would add a universal quantifier *for all* ∀ to `Eukaryota`, for example,

```
dataOntology:Eukaryota
    a owl:Class ;
    owl:equivalentClass [
        a owl:Class ;
        owl:intersectionOf ([
            a owl:Restriction ;
            owl:onProperty propertyOntology:
              has NucleatedCell ;
            owl:someValuesFrom dataOntology:
              NucleatedCell
          ] [
            a owl:Restriction ;
            owl:allValuesFrom dataOntology:
              Nucleated Cell > ;
            owl:onProperty propertyOntology:
              has NucleatedCell
        ])
  ].
```

In this case, only individuals that have one or more `hasNucleatedCell` property instances of which all values belong to the class `NucleatedCell` would be assigned to the class `Eukaryota`. Conversely, any individual assigned to the class `Eukaryota` is inferred to have at least one instance of the property `hasNucleatedCell`, and its value must be a datum belonging to the class `NucleatedCell` (even if that datum is not identified in the knowledge base).

OWL DL reasoning is designed under the open world assumption (OWA). As such, we cannot syntactically restrict the number of properties or the number of instances of a property. We can, however, achieve its logical equivalency with cardinality semantics. For example,

```
dataOntology:Nucleus
    a owl:Class ;
    rdfs:subClassOf [
        a owl:Restriction ;
        owl:cardinality "1"^^xsd:nonNegativeInteger ;
        owl:onProperty propertyOntology:hasNucleolus
    ].
```

The earlier discussion states that individuals that belong to the class `Nucleus` also belong to the class of all individuals that have at least one instance of the property `hasNucleolus`. The cardinality is 1, and thus if an individual has more than one instance of the property the reasoner will infer that the various objects that are the values of the property instances are semantically equivalent, that is, subject to an `owl:sameAs` relation connecting them. This will have cascading inference effects in the knowledge base. If other data contradict this, then the entire ontological model (knowledge base) is rendered inconsistent. Even if other data does not contradict this, inferring that two objects (the values of multiple `hasNucleolus` instances) are the same may yield surprising results. OWL DL's guarantees of completeness, validity, and monotonicity mean that factual errors cannot exist in a consistent knowledge base subject solely to first-order reasoning. But they may be uncovered when new data is added, a consequence that would lead to ontological inconsistency. This disruption is considered a good thing, because as more data is added and subject to a new round of ontology engineering, the resulting consistent ontology is a growing systematic endorsement of a true and consistent model of nature, whereas an inconsistent ontology forces resolution of the conflicting statements. Thus, there is a characteristic of convergence toward truth—a characteristic that is shared by the scientific method, even if it is only imperfectly realized.

Once the formal OWL ontology with named classes is constructed, we can subject it to tests for redundancy (two or more named classes inferred to be related by `owl:equivalentClass`), satisfiability (no class is equivalent to `owl:Nothing`), and closure (all data is realized to at least one class). We can use the formal ontology to the exclusion of the subsumption lattice, or we can use it as a marginal conceptualization on a subset of data. If the ontology includes necessary and sufficient conditions, we can use it to annotate new data or use new data to drive a new subsumption lattice to drive a new ontology.

Procedurally, the subsumption lattice drives the initial candidate classes to name and define; subsequent modeling allows us to derive formal named classes based on observed property conditions. If we do not axiomatically create classes de novo and assign individuals ex situ of the data, then we practice pure a posteriori, data-driven engineering. There are cases where we may value a priori knowledge, especially in cases of a priori *synthetic* statements in the Kantian sense (Kant 1787). For example, metadata and higher order logics may lead us to propose axiomatic classes that exceed the expressivity of OWL DL. In these cases, we pursue a *hybrid* of a priori and a posteriori engineering.

10.4.5 World Is Not First Order

An important caveat is that the success of this approach relies on the power and applicability of first-order description logics. First-order description logics are more expressive than the deep, static subsumption hierarchies so common in biology today. But our broader conceptualization of the world is not first order: *understanding* is reliant on higher order relations where the world cannot be neatly demarcated into mutually exclusive and exhaustive entities of individuals, properties, and classes. Even when given first-order constraints, OWL 1.1 DL has a complexity of NEXPTIME complete [NEXPTIME stands for nondeterministic exponential time: $O(2^{p(n)})$] (Cuenca Grau 2007), which is "harder" than NP complete (NP stands for nondeterministic polynomial time). OWL 2 DL is exponentially harder still (N2EXPTIME) (Kazakov 2008). Language-weakening constructs such as OWL 2 EL and OWL 2 QL can guarantee PTIME (polynomial time) complexity, but this comes at a cost (e.g., removing expressivity for the inverse relation or the universal quantifier *for all* ∀). Thus, we do not naively claim that FCA + OWL DL a posteriori ontology engineering is sufficient for all knowledge representation; rather, we offer it as an approach with quantifiable tractability for data-driven science.

10.4.6 Making Ontologies Web Savvy

SSWAP (Gessler et al. 2009) uses OWL ontologies as the foundation of a semantic web services platform. SSWAP supports the refactoring of

ontologies into terms with separately dereferenceable URIs, such that dereferencing terms returns OWL DL statements about each term.* Current research is moving in two directions: (1) delivering on guarantees on completeness (del Vescovo et al. 2011), and (2) establishing a resolver service so that the original ontologies will not need to be separately refactored.

SSWAP is the underlying technology for The iPlant Collaborative's semantic web platform.†,‡ It addresses limitations discussed in Section 10.2.3 by using ontology terms as resolvable URIs under a semantic web services protocol. Thus, ontologies—the conceptual constructs that describe data—are made instrumental in service description, discovery, invocation, and response. Semantic web services become the engagement layer for the underlying data and transformations upon them.

iPlant's semantic web platform uses transaction-time reasoning to match data types with service requirements, thereby allowing the construction of semantic pipelines of one service to the next. This matching is achieved with OWL reasoning. The entities of reasoning are service descriptions and their data, which are described with biological (or any third-party) ontologies. Thus, biological ontologies are being brought to bear on semantic web services a very real production environment. The missing link is that these biological ontologies are not themselves a posteriori reflections on data but exist as a priori knowledge management constructs. This makes data annotation burdensome and error prone, thus hindering the recruitment of data into the larger, semantic web service framework.

10.5 CONCLUSIONS AND THE PATH FORWARD

"Prediction is very difficult, especially about the future" (attributed to Niels Bohr [1885–1962]). We do not know how knowledge representation will evolve. We do know that current practices have limitations, and we discuss an approach to better formalize and quantify scientifically driven knowledge representation. Data-driven, a posteriori ontology engineering does not replace the a priori ontological construction of concepts from the empirical realization of individuals to classes (annotation). Rather, it advocates using measureable properties on individuals to simultaneously drive

* http://sswapmeet.sswap.info.
† http://www.iplantcollaborative.org/discover/semantic-web.
‡ http://sswap.info.

the concept space, its topology, and its realization. Challenges remain, in the implementation of FCA, in its mapping to FOL, and in the ability of FOL DL to suitably capture the relevant facts and relations. Yet the basic position of letting the data dictate the knowledge representation has promising characteristics. The most fundamental is a computationally tractable monotonicity, such that by adding more data we become increasingly confident that the resultant model gains comprehensive breath. It is a characteristic of the scientific endeavor—the pursuit, explicit or otherwise—of unifying theories, of a model or set of models that are internally consistent and that together cover the data-driven concept space with an explanatory and predictive power.

A litmus test for any knowledge representation is the degree to which it enables actionable, evidence-based decision making. Data-driven, a posteriori ontology engineering attempts to enable this by allowing the data to define the conceptualization, rather than vice versa. One may posit that when data are rare, deduction is underpowered and induction is a necessary risk. Today, especially in biology, we are entering the age of commoditization of data generation: data are not rare. This is especially evident in areas such as high-throughput DNA sequencing (e.g., see the study by Nowrousian [2010]). With data, deduction gains power, whereas induction and abduction (positing explanatory hypotheses based on informed analysis of the data) gain focus. Thus, data-driven, a posteriori ontology engineering is aimed at extracting the signal from the data, utilizing scientific data and measurements on them to drive a new understanding of the data's interconnectedness and ultimately their integration.

ACKNOWLEDGMENTS

This chapter is based on the work supported by the National Science Foundation (NSF) under grant #0943879 and the NSF Plant Cyberinfrastructure Program (#EF-0735191).

ABBREVIATIONS

DL:	description logic
FCA:	formal concept analysis
FOL:	first-order logic
GO:	Gene Ontology
IRI:	international resource identifier
LC:	Library of Congress
MeSH:	Medical Subject Headings
ORF:	open reading frame
OWL:	web ontology language
REST:	representational state transfer
RDF:	resource description framework
RDFS:	RDF schema
SSWAP:	simple semantic web architecture and protocol
UMLS:	Unified Medical Language System
URI:	uniform resource identifier
URL:	uniform resource locator
W3C:	World Wide Web Consortium
XML:	extensible markup language
XSD:	XML schema definition

REFERENCES

Akand E, Bain M, Temple M. 2007. Learning from ontological annotation: An application of formal concept analysis to feature construction in the gene ontology. In: *Proceedings of the Third Australasian Ontology Workshop* (AOW-2007), Gold Coast, Australia. CRPIT eds. T. Meyer and A. Nayak, Vol. 85, 15–23.

Akand E, Bain M, Temple M. 2010. Learning with gene ontology annotation using feature selection and construction. *Applied Artificial Intelligence*, 24:5–38.

Alterovitz G, Xiang M, Hill DP, Lomax J, Liu J, Cherkassky M, Dreyfuss J, Mungall C, Harris MA, Dolan ME, Blake JA, Ramoni MF. February 2010. Ontology engineering. *Nature Biotechnology*, 28(2):128–130. PMID: 20139945.

Antezana E, Egaña M, Blondé W, Illarramendi A, Bilbao I, De Baets B, Stevens R, Mironov V, Kuiper M. 2009. The cell cycle ontology: An application ontology for the representation and integrated analysis of the cell cycle process. *Genome Biology*, 10:R58. doi:10.1186/gb-2009-10-5-r58.

Aristole. 350 BCE. On the Parts of Animals. Trans. by William Ogle. Kegan Paul, Trench & Co., London, 1882.

Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, et al. 2000. Gene ontology: Tool for the unification biology. Gene Ontology Consortium. *Nature Genetics*, 25(1):25–29.

Bada M, Livingston K, Hunter L. 2011. An ontological representation of biomedical data sources and records. In: *Proceedings Bio-Ontologies SIG at ISMB 2011*, Vienna, Austria, 75–78.

Bain M. 2002. Structured features from concept lattices for unsupervised learning and classification. In: *AI 2002: Proceedings of the 15th Australian Joint Conference on Artificial Intelligence*, eds. B. McKay and J. Slaney, LNAI 2557, 557–568, Berlin: Springer.

Bain M. 2003. Inductive construction of ontologies from Formal Concept Analysis. In: AI 2003: Advances in Artificial Intelligence. 16th Australian Conference on AI, Perth, Australia, December 3–5. Lecture Notes in Computer Science Volume 2903, http://link.springer.com/chapter/10.1007%2F978-3-540-24581-0_8, 88–99.

Belleau F, Nolin MA, Tourigny N, Rigault P, Morissette J. 2008. Bio2RDF: Towards a mashup to build bioinformatics knowledge systems. *Journal of Biomedical Informatics*, 41:706–716.

Cimiano P, Hotho A, Staab, S. 2005. Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. *Journal of Artificial Intelligence Research*, 24:305–339.

Corcho O, Fernández-López M, Gómez-Pérez A. 2007. Ontological Engineering: What are Ontologies and How Can We Build Them? In: *Semantic Web Services: Theory, Tools and Applications* IGI Global, ed. J Cardoso, 44–70.

Cuenca Grau, B 2007. OWL 1.1 Web Ontology Language. Tractable Fragments. http://www.webont.org/owl/1.1/tractable.html.

del Vescovo C, Gessler DDG, Klinov P, Parsia B, Sattler U, Schneider T, Winget A. 2011. Decomposition and modular structure of BioPortal ontologies. In: *Proceeding of International Semantic Web Conference ISWC 2011*, Bonn, Germany, LNCS 7031, 130–145.

Fernández-López M, Gómez-Pérez A, Juristo N. 1997. METHONTOLOGY: From Ontological Art Towards Ontological Engineering. Spring Symposium on Ontological Engineering of AAAI, 33–40, Stanford, CA: Stanford University.

Fernández-López M, Gómez-Pérez A. 2002. Overview and analysis of methodologies for building ontologies. *The Knowledge Engineering Review*, 17:129–156. doi:10.1017/S0269888902000462.

Fielding RT. 2000. Architectural styles and the design of network-based software architectures. Doctoral dissertation, University of California.

Formica A. 2006. Ontology-based concept similarity in Formal Concept Analysis. *Information Sciences*, 176:2624–2641.

Formica A. 2008. Concept similarity in Formal Concept Analysis: An information content approach. *Knowledge-Based Systems*, 21:80–87.

Ganter B, Stumme G, Wille R. eds. 2005. *Formal Concept Analysis: Foundations and Applications*, Springer-Verlag, Berlin Heidelberg.

Ganter B, Wille R. 1999. *Formal Concept Analysis*, Springer-Verlag, Berlin Heidelberg.

Gessler DDG, Schiltz GS, May GD, Avraham S, Town CD, Grant D, Nelson RT. 2009. SSWAP: A Simple Semantic Web Architecture and Protocol for semantic web services. *BMC Bioinformatics*, 10:309. doi:10.1186/1471-2105-10-309.

Gómez-Pérez A, Fernández-López M, Corcho, O. 2003. *Ontological Engineering*. Springer Verlag, Berlin Heidelberg.

Guo L, Huang F, Li Q, Zhang GQ. 2011. Power contexts and their concept lattices. *Discrete Mathematics*, 311(18–19):2049–2063.

Joslyn C, Paulson P, Verspoor KM. 2008. Exploiting term relations for semantic hierarchy construction. In: *Proceedings of the International Conference on Semantic Computing (ICSC 08)*, 42–49, IEEE Computer Society, Los Alamitos CA.

Kaiser T, Schmidt S, Joslyn C. 2008a. Concept lattice representations of annotated taxonomies. *Concept Lattices and their Applications, Lecture Notes in AI*, eds. S. B. Yahia, E. M. Nguifo, R. Belohlavek, Vol. 4923, 214–225, Berlin: Springer-Verlag.

Kaiser T, Schmidt S, Joslyn C. 2008b. Adjusting Annotated Taxonomies. *International Journal of Foundations of Computer Science*, 19(2):345–358.

Kant I. 1787. *Critique of Pure Reason*. Second edition. Translated by Norman Kemp Smith. England: Palgrave Macmillan.

Kazakov Y. 2008. RIQ and SROIQ are Harder than SHOIQ. In: *Proceedings of the 21st International Workshop on Description Logics* (DL2008), Dresden, Germany, May 13–16, 2008.

Linné, C. von. 1735. *Systema Naturæ per Regna Tria Naturæ, Secundum Classes, Ordines, Genera, Species, cum Characteribus Differentiis, Synonymis, Locis* [System of Nature, in Three Kingdoms of Nature, with Classes, Orders, Types and Species, with Differences of Character, Synonyms, Places] Joannes Wilhelm de Groot for Theodor Haak, Leiden.

McGuinness DL, van Harmelen F. 2004. OWL Web Ontology Language. Overview. http://www.w3.org/TR/owl-features.

Nowrousian M. September 2010. Next-generation sequencing techniques for eukaryotic microorganisms: Sequencing-based solutions to biological problems. *Eukaryotic Cell*, 9(9):1300–1310.

Simperl EPB, Tempich C. 2006. Ontology engineering: A reality check. R. Meersman, Z. Tari et al. (Eds.): OTM 2006, LNCS 4275, Springer-Verlag Berlin Heidelberg, 836–854.

Sjoblom T, Jones S, Wood LD, Parsons DW, Lin J, Barber TD, Mandelker D, Leary RJ, Ptak J, Silliman N, Szabo S, Buckhaults P, Farrell C, Meeh P, Markowitz SD, Willis J, Dawson D, Willson JK, Gazdar AF, Hartigan J, Wu L, Liu C, Parmigiani G, Park BH, Bachman KE, Papadopoulos N, Vogelstein B, Kinzler KW, Velculescu VE. 2006. The consensus coding sequences of human breast and colorectal cancers. *Science*, 314(5797):268–274. Epub 2006 Sep 7. PubMed ID: 16959974.

Stevens R, Aranguren ME, Wolstencroft K, Sattler U, Drummond N, Horridge M, Rector A. 2007. Using OWL to model biological knowledge. *International Journal of Human-Computer Studies*, 65:583–594.

Sure Y, Tempich C, Vrandecic D. 2006. Ontology engineering methodologies. In: *Semantic Web Technologies: Trends and Research in Ontology-based Systems*, 171–190. eds. J. Davies, R. Studer and P. Warren, Chichester, UK: John Wiley & Sons, Ltd.

Whittaker RH. 1969. New concepts of kingdoms of organisms. *Science*, 163: 150–161.

Woese CR, Balch WE, Magrum LJ, Fox GE, and Wolfe RS. 1977. An ancient divergence among the bacteria. *Journal of Molecular Evolution*, 9:305–311.

Woese CR, Kandler O, and Wheelis ML. 1990. Towards a natural system of organisms: Proposal for the domains Archaea, Bacteria, and Eucarya. *Proceedings of the National Academy of Sciences*, 87:4576–4579.

CHAPTER 11

Transforming Data into the Appropriate Context

Bill Howe

CONTENTS

11.1 Problem Overview 247
11.1.1 Long Tail Science 251
11.1.2 How Big Is Big? 252
11.1.3 Problem Summary 252
11.2 Vision for the Future: Computing with BAGS 253
11.2.1 Bottom-Up: Inclusive Rather Than Exclusive 254
11.2.2 Algebraic: Reasoning about Computation 256
11.2.3 Global: Universal Addressability 257
11.2.4 Speculative: Proactive Derivation of Data Products 259
11.3 Relationship to Other Technologies 261
11.3.1 Cloud Computing 261
11.3.2 Service-Oriented Architectures 262
11.3.3 Scientific Workflow 263
11.3.4 Databases 263
11.3.5 Dataspaces 264
11.3.6 Data Integration 264
11.3.7 Visualization 264
11.4 State of the Art 264
11.4.1 SQLShare: Smart Services for Ad Hoc Databases 264
11.4.1.1 Data Model 267
11.4.1.2 Automatic Starter Queries 268
11.4.2 Self-Organizing Dashboards 271