Towards a Semantic Knowledge Base for Yeast Biologists

Natalia Villanueva-Rosales
School of Computer Science
Carleton University
+1-613-520-2600 x4491
nvillanu@scs.carleton.ca

Kevin Osbahr
Department of Biology
Carleton University
+1-613-520-2600 x4491
nis_pero@hotmail.com

Michel Dumontier
Department of Biology, Institute of Biochemistry, School of Computer Science
Carleton University
1125 Colonel By Drive, Ottawa, Canada K1S 5B6
+1-613-520-2600 x4194
michel_dumontier@carleton.ca

ABSTRACT
The integration of data from heterogeneous sources is an ongoing challenge for the scientific community. The semantic web initiative provides a new knowledge engineering framework to represent, query and share information. In this paper, we describe our efforts towards the development of an ontology-driven knowledge base that allows semantic query answering of yeast knowledge.

Categories and Subject Descriptors
H.1 [Models and Principles]: OWL ontology
I.2.4 [Knowledge Representation Formalisms and Methods]:
J.2 [Physical Sciences and Engineering]
J.3 [Life and Medical Sciences]: Biology and genetics

General Terms
Algorithms, Management, Design, Standardization, Languages.

Keywords
Semantic Web, data integration, semantic query answering, knowledge management, OWL, ontology, biological data, yeast, Saccharomyces cerevisiae.

1. INTRODUCTION
Online biological information is available via web pages, stored in databases and described in publications. However, web search engines are unable to find information with a set of specific properties. The problem is that the *representation* of biological information on the web *is not machine understandable*, in the sense that computers cannot interpret words, sentences, so as to correctly reason about the objects such words represent and the relations between them that are implicitly stated in those sentences [1]. The primary goal of the *semantic web* is to add *semantics* to the current Web, by designing *ontologies* which explicitly describe and relate objects using formal, logic-based representations that a machine can understand and process [2, 3]. This ongoing effort is expected to facilitate data integration and semantic querying of knowledge, of critical importance in the life sciences.

WWW 2007, May 8--12, 2007, Banff, Canada.

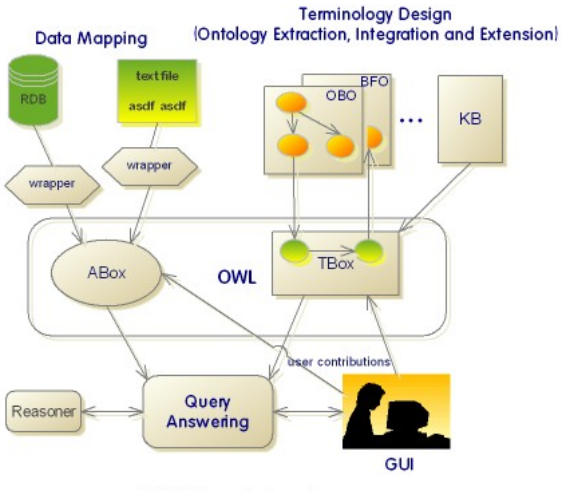

Figure 1 yOWL System Overview

Ontologies already play an important role in managing medical terminology [4-6], and more recently in the discovery and execution of grid [7] and semantic web services [8]. The Open Biomedical Ontologies (OBO) is a shared portal of biological/medical ontologies that includes the popular Gene Ontology (GO) [9]. By providing a standardized vocabulary, OBO controlled vocabularies and taxonomies are used in the annotation of biological information, which helps make information more accessible for computer interpretation. Through the OBO Foundry effort, OBO ontologies are being redesigned and mapped to the Basic Formal Ontology (BFO) [10], an ontology that provides distinction between objects and processes and can be linked using basic relations [11]. Together, they should provide a powerful platform to describe and annotate domain specific knowledge, and open the possibility of making queries at various levels of granularity and moreover, queries that retrieve information from diverse domains. For instance, a biochemical ontology might state that enzymes are types of proteins that catalyze reactions, and this information would facilitate querying a knowledge base using the term “protein” to retrieve all data that has been annotated as a protein that catalyzes some reaction or an enzyme. Despite the OBO Foundry effort, OBO ontologies cannot be used in this way because they do not contain explicit logical descriptions to define class membership in terms of their properties. For instance, you would not be able to query a database about individuals that are proteins and catalyze

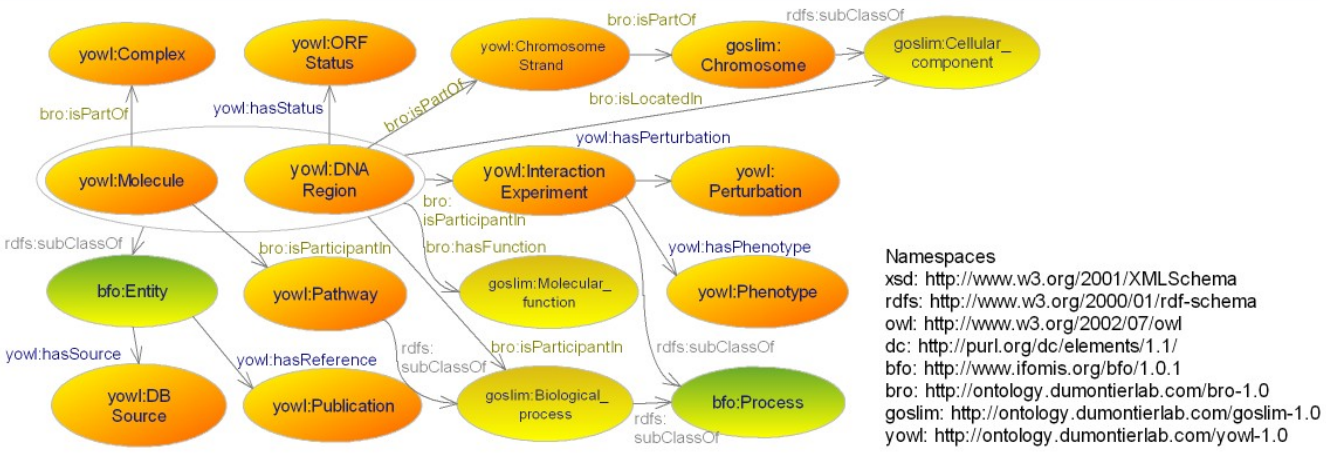

Figure 2 Select classes and relations from yOWL ontology

reactions based on an OBO ontology. Therefore, the next step is to make explicit the semantics for OBO ontologies by using formal, logic-based knowledge representation languages. Related work towards this goal can be found in [12].

OWL, the Web Ontology Language [13], is the official recommendation to create semantic web ontologies and is a knowledge representation language with in which one variant, OWL-DL, is based on description logics (DL), a subset of First Order Logic that allows description of complex concepts from simpler ones with an emphasis on decidability of reasoning tasks [14]. Reasoning tasks like checking ontology *consistency*, computing *inferences*, and *realization* (classifying real world objects into their most specific category) can be executed by a computer program called a reasoner (e.g. Pellet [15] and Racer Pro [16]) over DL ontologies [17]. The design of OWL-DL bio-ontologies favorable to reasoning may be achieved by the application of semantic web best practices [18], relation formalisms [11], normalization [19], design patterns and workflows [20]. Sophisticated biomedical ontologies such as the Foundational Model of Anatomy are being converted to OWL and this has proven useful in simplifying the ontology and identifying inconsistencies [21, 22]. The FungalWeb project involved the design of an ontology to reason about enzymes of importance to the yeast biotechnology industry [23]. The BioPax OWL ontology [24] provides a simple ontology to represent pathways, interactions and molecular participants, which has been used by pathway data providers such as HumanCyc [25] and Reactome [26] to share knowledge. In BioPax, data integration occurs by instantiating classes for cross references, rather than using the semantics provided by OWL.

Indeed, a major challenge in bioinformatics is the cross-referencing of the overwhelming number of identifiers for biological data that refers to the same entities. The proliferation of identifiers stems from 1) direct user submissions to a specific database, 2) the import of data into "boutique" databases and 3) value added annotations fuels a need for each provider to issue new identifiers so as to keep track of their contributions [27-29]. However, keeping track of these identifiers is such a problem that it becomes necessary to create databases of database identifiers [30, 31]. In fact, identifiers have such a pervasive influence in the life sciences that people talk about identifiers instead of the entities they are meant to identify. While LinkHub [32] provides a first step at navigating this confusing set of identifiers, YeastHub [33] provides an RDF-based data warehouse which lets one add data and create queries between resources. While flexible, the lack of an ontology requires the user to indicate which user-contributed resources are equivalent and this precludes the automatic discovery of semantically equivalent knowledge.

In this paper, we present a first approach to represent knowledge found in the Saccharomyces Genomes Database [34] using an OWL ontology that extends the BFO. We demonstrate the utility of this approach for resolving issues surrounding multiple resource identifiers and demonstrate how this ontology may be used to guide the construction of sophisticated, semantically correct queries that can be answered by a reasoner for knowledge discovery.

2. METHODS

The yOWL system overview is shown in Figure 1. yOWL is comprised of three major components: ontology design, data mapping and query answering.

2.1 Ontology Design

All ontologies were designed using Protégé (v 4.29 alpha).

The yOWL ontology was manually created and designed following the semantic web best practices [11, 18-20] using OWL-DL. The goal of the yOWL design was to cover the entities and their relations modeled in the SGD relational database available at http://www.yeastgenome.org/, which is the source of our data. Table 1 lists the data obtained from Saccharomyces Genome Database (SGD). This data includes structural and functional chromosome features (telomeres, genes, etc), database cross references, molecular function, cellular component, biological process, interactions, pathways, phenotypes and literature references. These concepts were mapped to the OWL-DL version of BFO which provides disjunction between qualities, functions, roles, objects, object parts, processes and spatial and temporal regions.

Basic relations between entities described in [10, 11] were incorporated in an OWL-DL ontology termed the Basic Relation Ontology (BRO), available at http://ontology.dumontierlab.com/bro. The BRO is hierarchically organized from a root relation "isRelatedTo" to groups of object-process, parthood, spatial and temporal relations. Next, the BRO's domain and range values were mapped to classes in the Basic Formal Ontology (BFO), resulting in an integrated upper level ontology termed NULO, available at http://ontology.dumontierlab.com/nulo. NULO consists of 36 classes, 50 object properties and 2 datatype properties, 17 annotation properties. With domain and range assignments, NULO provides the BFO with relations that are constrained in a semantically correct manner.

Classes: The classes of yOWL were initially extracted from the attributes in the flat files and later on they were augmented and refined to reflect knowledge about genome structure and function. For instance, the interactions file contains experimental data about interactions. It has the attributes: orf1, orf2, interaction_type, viability and pmid. From this file, we created the following classes: i) The class InteractionExperiment plus the set of subclasses described in the interaction_type (e.g. SyntheticLethality, AffinityCapture-MS, etc). With domain knowledge, the subclasses of InteractionExperiment were grouped in two main classes: PhysicalInteractionExperiment and GeneticInteractionExperiment. Finally, an Experiment class was added as a parent class of InteractionExperiment and as a subclass of the BFO Process class for ontology integration. ii) Classes to represent the viability types, included in the viability attribute were created using the same criteria iii) The class Publication was created as a subclass of the BFO Object class to represent the attribute pmid. Notice that the attributes orf1 and orf1 refer to instances of Open Reading Frames, but this class was created from the SGD_features file, and therefore will only be related using an object property. Finally, class definitions were obtained from WordNet and the SGD glossary and were added to the class using the "comment" annotation property.

Object properties: New object properties were added to describe the more specific relations required (but not restricted to) in this domain. The first ("hasReference"/ "isReferencedIn"), provides a relation between a publication and the entity it references. The second ("hasSource"/"isSourceOf"), links an entity with its source of origin, and provides a means to assign data provenance. A quality relation ("hasStatus"/ "isStatusOf") describes the status (verified, dubious, uncharacterized) of an open reading frame. Finally, two object relations ("hasOutcome"/ "isOutcomeOf" and "hasPerturbation"/ "isPerturbationOf") describe the relationship between an entity and an outcome (such as a phenotype) and a perturbation (i.e. deletion of a gene), respectively. Ideally, these properties would be associated with the participants of some experiment, but given the form of the data, they are associated directly with experiments.

Data properties: Several data properties were also introduced to accommodate information that is intrinsic to the specific entity. For instance, the date version properties (sequence version and coordinate version), in which multiple values should be maintained. Most chromosomal features are associated with a start and end coordinate which delineate a continuous region located on chromosomal strand. Another property provides the association of a citation with a publication ("hasCitation"). Finally, a data property was established to relate a gene with the biochemical reaction that its gene product catalyzes ("hasECNumber"), although it is expected that this relation will later be converted to one between a protein product and a biochemical reaction.

The ontology (shown in part in Figure 2) extends NULO so as differentiate between processes (pathways, experiments) and relates them to their participant objects (i.e. DNA). To ensure the proper classification of data and knowledge discovery, necessary and necessary and sufficient conditions were added to the ontology. Necessary conditions were added where obvious (i.e. a chromosome strand is a single stranded DNA molecule that is part of a chromosome). Necessary and sufficient conditions were added for single property varying entities (i.e. a dosage rescue experiment is a dosage interaction experiment that has a viable outcome) or in the design of a value partition (i.e. the quality of viability is defined by a state of viable or non-viable).

The yOWL ontology, including BFO and GOSLIM is currently comprised of 244 classes, from which 38 are defined classes, 57 object properties, 11 data properties and 22 annotation properties. It is available at http://ontology.dumontierlab.com/yOWL-1.0 .

2.2 Data Mapping

The processing of SGD data was a significant challenge, given that it is obtained from a non-normalized database schema. We will now describe how we overcome the challenges of the data mapping. Tab-delimited files used for this study are listed in Table 1. Normalization of certain files (complex, interactions) was necessary as multi-valued entries were separated by "/" or "|" delimiters. Another problem encountered was that many files did not use the SGD identifier (SGDID; a numeric identifier prefixed with an 'S' – chromosomal features file) as a foreign key, but instead contained references to gene names or ORF names. The data was imported into the yOWL ontology using PHP-based text2owl parsers that we developed in house. The data was assigned to the namespaces shown in Table 2.

Table 1. Resources added to yeast knowledge base

Data Type	***File***	***Number of records***
Chromosome features	SGD_features.tab [1]	16,781
DB Cross References	dbxref.tab [1]	68,313
Function, localization, process	go_slim_mapping.tab [2]	21,176
Interactions	Interactions.tab [2]	148,777
Complex	go_protein_complex_slim.tab [2]	3,105
Pathways	biochemical_pathways.tab [2]	946
Phenotypes	phenotypes.tab [2]	15,505
Literature	gene_literature.tab [2]	142,777

[1] ftp://genome-ftp.stanford.edu/pub/yeast/chromosomal_feature/

[2] ftp://genome-ftp.stanford.edu/pub/yeast/literature_curation/

Table 2. Resource Namespaces

Source	*URI*
yOWL ontology	http://ontology.dumontierlab.com/yowl-1.0
GOslim ontology	http://ontology.dumontierlab.com/goslim-1.0
GO	http://geneontology.org/go
SGD[1]	urn:lsid:yeastgenome.org:
Genbank	urn:lsid:ncbi.nlm.nih.gov:genbank:
PubMed	urn:lsid:ncbi.nlm.nih.gov:pubmed:
EBI	urn:lsid:ebi.ac.uk:
DIP[1]	urn:lsid:dip.doe-mbi.ucla.edu:
CGD[1]	urn:lsid:candidagenome.org:
CandidaDB[1]	urn:lsid:candidadb:
IUBMB[1]	urn:lsid:iubmb:reaction:
EUROSCARF[1]	urn:lsid:euroscarf:
BioGrid[1]	urn:lsid:thebiogrid.org:
MetaCyc	urn:lsid:metacyc.org:
GermOnline[1]	urn:lsid:germonline.org:

[1]LSID assigned in the absence of known authority.

Using an Intel Pentium 4 computer with 4GB RAM, we were able to load the entire data-instantiated ontology using Protegé 4, but unable to enable on the reasoner for query answering. We also attempted to load the ontology with Racer, but were unable to query the ontology due to time and resource restrictions. Thus, due to issues with reasoning performance, the query examples described in this paper were selected from a subset of the full yOWL instance data (ABox). This subset was obtained by performing a database search for genes with links across the ontology. While limited, this approach still enabled a demonstrated of proof of principle on how OWL-DL ontologies can be used to enable semantic query answering over data, rather than undertake performance testing on available tools. The resulting ABox subset contained 3,423 instances.

2.3 Query answering

The design and population of ontologies and the use of reasoning capable applications will aid researchers to retrieve specific information and discover new relations about their subject of interest. Our goal is to show by means of examples how a scientist could extract information from yOWL, identify equivalent class definitions to the query is being posed, query at various levels of granularity and across data sources solving the problem of multiple identifiers. We focus on two main categories of queries: class queries and graph pattern based queries.

Class queries are useful when the goal is to retrieve a set of individuals that satisfy certain restrictions. These restrictions, logically describe the membership requirements for an individual to belong to a class. For instance, *find all individuals that are located in the mitochondrion*. In this query, the restriction is that the individual is located in the mitochondrion. Class queries are equivalent to defined class descriptions and therefore can also be used to retrieve superclasses, subclasses and equivalent classes of the class being defined. The formulation of a query is constrained to the entities, relations and individuals defined by the ontology. These queries were formulated using the Manchester OWL syntax [35]. OWL class queries were performed using the Protégé 4 DL Query plugin and the Pellet reasoner that is embedded in this application.

When the goal is not only to identify a set of individuals that belong to a described class, but also to identify the other individuals that each member of this class is related to in order to satisfy the class membership restrictions, a more expressive query language is needed. For example, *find the set of identifiers and their database sources for genes (gene products) involved in a protein modification pathway*. Here, the query should retrieve not only the set of genes that satisfy the restriction, but also their identifiers and database sources. A more expressive query can be formulated using variables in the restrictions (i.e. the variables that will be instantiated with those individuals that satisfy the query). These type of queries can formulated using nRQL query language, a lisp based query language [16] supported in Racer Porter v.1.9.0 and Racer Pro. We called these queries graph pattern based queries. Intuitively, these queries are based on graph patterns composed by nodes and edges. Each node can represent an unbound variable that will be bound to a member of a certain class or a variable already bound to a specific (named) individual. Edges represent relations through properties or restrictions in the ontology. The mapping from the graph pattern to the nRQL query was manually done. For nRQL query answering we used Racer Porter applying unique name assumption, negation as failure and indexation.

3. RESULTS

3.1 Heterogeneous Data Integration

3.1.1 Resource integration and provenance

SGD assigns a unique identifier (SGDID) for every chromosomal feature it provides. All other identifiers including gene names, gene aliases, ORF names, and all database cross-references were assigned one of the namespaces in Table 2 and made an instance of bfo:Entity. Identifiers that point to the same resource were made equivalent by asserting the OWL "sameAs" relationship to the SGD identified resource. This enables a DL reasoner to infer that all database cross references or identifiers point to the same resource. Therefore, statements made using any one of the various identifiers would automatically be resolved and this would greatly simplify the data import process. In addition, provenance is maintained by assigning the LSID namespace to imported data, as well as linking resources to their databases with a "hasSource" object property. This then enables querying data from a specific source either by filtering namespaces or by querying object properties.

3.1.2 Instantiation of the Gene Ontology

By adopting the BFO, we subscribe to the idea that molecular functions, cellular components and biological processes really do exist in the real world and as such are instances of their respective classes. For simplicity and to reduce the complexity of reasoning, we designed an OWL ontology with 79 classes spanning the 3 hierarchies (molecular function, cellular component and biological process) based on the yeast GO slim collection available at http://www.yeastgenome.org/. The OWL-DL GO slim ontology is available at

http://ontology.dumontierlab.com/goslim). We used the GO slim ontology rather than the full GO ontology (with over 19,000 terms). We created an instance of the correspondent GO term for each GO annotated gene/protein, which opens the door to making future statements about those particular functions, processes and components. For backwards compatibility, each entity is also linked to a generic instance of the ontology named by the GO identifier.

3.2 Semantic Query Answering

3.2.1 Types of queries posed to yOWL

Table 3 lists some examples for the two types of queries that demonstrate the basic functionality and advantages of ontology-driven queries to the yeast biologist.

Table 3. Example queries posed to yOWL

Type	*Query*	*Query Feature*
Class Queries	1. Find all individuals that have a molecular function.	Existential restriction, ontology integration.
	2. Find all uncharacterized open reading frames that have a known molecular function	Conjunctive query, ontology integration.
	3. Find pathway participants that are also physical interaction participants.	Conjunctive query, defined classes.
	4. Find all open reading frames on chromosome 5.	Transitive relations
	5. Find all the interaction experiments that are referred in at least 4 publications.	Cardinality restrictions
	6. Find all DNA regions that are not physically mapped.	Negation, defined classes
Graph Pattern based Queries	7. Find genes that play a role in transcription and are participants in some genetic interaction experiment. Return genes and their publications.	Conjunctive queries, variable binding
	8. Give the set of identifiers and their database sources for genes involved in a protein modification pathway.	DB cross references, variable binding
	9. Find all information related to Gene NSA3	Property/role hierarchy
	10. Find genes/proteins with transferase activity that are part of a complex, have rescued a non-viable phenotype by overexpression and have a known role in some pathway. Retrieve genes, sources, pathways, experiments, chromosome and complexes	Conjunctive query, variable binding, ontology integration.

3.2.2 Query Results

Query 1. Find all individuals that have a molecular function.

Class Expression: hasFunction some Molecular_function

This query returns the set of individuals that have some (at least one) known molecular function. In order to answer this query, the GO slim ontology has to be integrated. Among the results we find the individual with SGD identifier S000005174, described as an Elongin A, F-box protein. If we search in our knowledge base, this individual has been associated with an instance of TranscriptionRegulatorActivity named 'S000005174_GO_Transcription_regulator_activity'. This simple query can be used as a building block for more sophisticated queries.

Query 2. Find all uncharacterized open reading frames that have a known molecular function.

Class Expression: OpenReadingFrame that (hasStatus some {uncharacterized}) and (hasFunction some Molecular Function)

This query illustrates the imposition of multiple restrictions on an individual. The set of conditions (known molecular function, uncharacterized ORF) are joined by conjunction and are therefore called *conjunctive queries*. The result to this query contains among others, an individual with the SGD identifier S000005255 that is characterized as a putative F-box protein and has protein binding as a molecular function. Since uncharacterized open reading frames are those that likely encode a protein, but for which there are no specific experimental data demonstrating that a gene product is produced in *S. cerevisiae*, such queries open new avenues for experimental investigation and validation.

Query 3. Find pathway participants that are also physical interaction participants.

Class Expression: PathwayParticipant and PhysicalInteractionParticipant

This query illustrates the use of conjunctive queries with *defined classes* for knowledge discovery. A defined class relies on necessary and sufficient conditions to logically describe its membership requirements. An OWL-DL reasoner (e.g. Pellet, Racer) will discover which individuals satisfy the class restrictions and will classify such individuals as instances of that defined class. The yOWL ontology contains the class PathwayParticipant for which the necessary and sufficient conditions for membership are: i) be an instance of an independent continuant and ii) is a participant in some pathway. The class InteractionParticipant is defined to be i) an instance of an independent continuant and ii) participant in some physical interaction experiment. The full query can be posed in terms of primitive (not defined) classes: *Find all continuants that participate in a pathway and participate in a physical interaction experiment.* Both queries return the same set of individuals as an answer, which includes among others: the YJL031C with SGD identifier S000003568. In fact, this protein has a known role in a protein modification pathway and has been shown to interact in five physical interaction experiments (affinityCapture-MS, FRET, reconstituted complex , dosage rescue and co-purification). Thus, knowledge that spans different curated information can be easily queried.

Query 4. Find all open reading frames (ORF) on chromosome 5.

Class Expression: OpenReadingFrame that isPartOf value chromosome5

Open reading frames are part of chromosome strands which themselves are part of a chromosome. The result of this query includes an individual with SGD identifier S000002954 that corresponds to the YEL059C-A ORF. However, in our knowledge base, it is only *asserted* that this ORF is part of the Crick strand that is part of chromosome 5. Given that the "part of" relation is *transitive*, the reasoner can infer *that* this ORF is also part of the chromosome 5.

Query 5. Find all the interaction experiments that are referred in at least 4 publications.

Class Expression: InteractionExperiment that (hasReference min 4 Publication)

This query imposes cardinality restrictions over the property "has Reference". The result contains the AffinityCapture-MS interaction experiment with SGD identifier interaction_173, which is referred in the publications identified by PMID 1805837, 12374754, 16429126 and 16554755. In Racer, queries including cardinality restrictions can not contain transitive relations [36]. Moreover, when the cardinality restrictions involve operators like "at most" or "exactly", it will be necessary to "close the world" at query time, and it should be interpreted more as has at most 4 references known that has at most 4 references. Also notice that for this type of queries, the "unique name assumption" has to be turned on for query answering purposes. This configuration option is offered by both, Protégé and Pellet reasoners. Otherwise, axioms containing the OWL "differentFrom" property axioms would need to be added to the ontology.

Query 6. Find all DNA regions that are not physically mapped.

Class Expression: DNARegion and not (hasStartCoordinate some int) and not (hasEndCoordinate some int)

DNA Regions that are physically mapped have a known start coordinate and end coordinate along the chromosome. The result of this query contains, among others, the individual with SGD identifier S000029174, a negative regulator gene in general amino acid biosynthetic pathway. This query also matches the description of the defined class "NotPhysicallyMappedFeature" in the ontology. Notice that for this type of queries, we used the Negation as Failure provided by nRQL in Racer. The answers of these queries should be interpreted as a "not known" (at the time of query) about evidence to support a true statement, and therefore is considered as false, or rather, not known to be true.

The following queries are considered under the category of graph pattern based query, and therefore, the result will contain the set of values (bindings) for each variable (node) in the pattern graph queried that satisfy the conditions described in such a graph.

Query 7. Find genes that play a role in transcription and are participants in some genetic interaction experiment. Return both the genes and their associated publications.

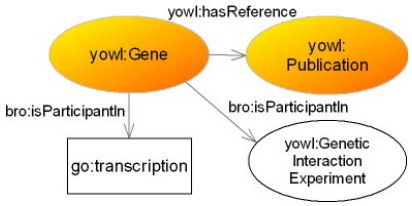

Figure 3 Graph pattern for Query 7. Variables included in the answer (filled circle), variables not returned in the answer (unfilled circle), individuals (square).

The result of this query is a set of tuples, each one containing the values for publications and the genes they reference that satisfy both conditions: i) play a role in transcription and ii) are participant in some genetic interaction experiment. This result contains among others the tuple: publication (PMID:16431986) with the gene (S000005705) whose gene product is a part of the APT subcomplex of cleavage and polyadenylation factor.

Query 8. Give the set of identifiers and their database sources for genes involved in a protein modification pathway.

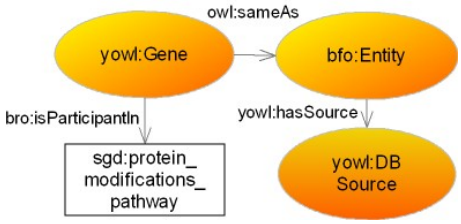

Figure 4 Graph pattern for Query 8. Variables included in the answer (filled circle), individuals (square).

This query illustrates the use of OWL "sameAs" property to cross reference identifiers of the same real world object to integrate information about this object described in heterogeneous systems. The result of this query contains (among others) the individual with the identifiers and sources (id from source): S000003568, YJL031C and BET4 from SGD, PWY30-11 from MetaCyc, YJL031C (also found as YJL031c) from EUROSCARF, 4994 from DIP, orf10.1039 from CGD, CA1034 from CandidaDB, CAA89323.1 and AAA21386.1 from GenBank/EMBL/DDBJ, 33728 from BioGRID, UPI000034F5CE and Q00618 from EBI, NP_012503.2 and 853421 from NCBI.

Notice that in some sources, the same individual has more than one identifier. In some systems, data curation is needed to match YJL031c and YJL031C. In yOWL however, the information related with this identifier, is integrated given that they are refer to the same object, which is a more accurate representation of the real world.

Query 9. Find all information related to Gene NSA3.

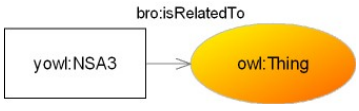

Figure 5 Graph pattern for Query 9. Variables included in the answer (filled circle), individuals (square).

As mentioned in section 2.1, yOWL contains a property hierarchy, whose top property is "isRelatedTo". Therefore, every relation (asserted or inferred) between NSA3 and any other individual, will imply that NSA3 is related to that individual. This query retrieves all the individuals NSA3 is related to at the most general level of granularity: (sources) EBI, CGD, NCBI, GenBank/EMBL/DDBJ, BioGRID, DIP and CandidaDB, (ORFStatus) verified, chromosome8_Watson, proteasome_complex, (GO identifiers) GO_S000001094_Ribosome_biogenesis_and_assembly, GO_S000001094_Nucleolus, GO_S000001094_Protein_catabolic_process and GO_S000001094_Protein_binding. It is also related to a large set of interactions, and a set of publications including PMID:16922378. This exploratory query can later be refined searching for more specific types of relations between NSA3 and other individuals (e.g. find all the molecular functions related with NSA3 or the location of NSA3).

Query 10. Find genes/proteins with transferase activity that are part of a complex, have rescued a non-viable phenotype by overexpression and have a known role in some pathway. Retrieve the genes, sources, pathways, experiments, chromosome and complex that satisfy these requirements.

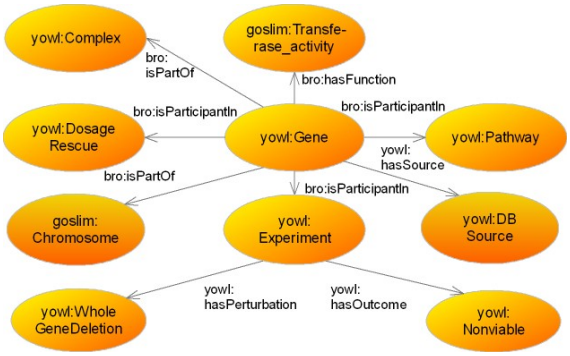

Figure 6 Graph pattern for Query 10. Variables included in the answer (filled circle).

This query represents a more sophisticated query that a yeast scientist might pose. It requires the integration of all the information including ontology-based inferences (e.g. be part of a chromosome) and ontology integration (e.g. the transferase activity GO term). The answer to this query is the set of tuples containing a gene, source, pathway, experiment, chromosome and complex that together with the relations satisfy the restrictions defined in the query. An answer to this query is the gene BET4, with source SGD that is a participant in the pathway_1 and also in the experiment_30. This experiment has as outcome the phenotype_30 that has the quality of being non-viable. BET4 also participates in the interaction_871 obtained from a Dosage Rescue Genetic Interaction Experiment. This gene is annotated with the S000003568_GO_Transferase_activity, an instance of the GO transferase activity molecular function and is also part of the chromosome10 and the Rab-protein_geranylgeranyltransferase_complex, which is an instance of complex. Each unbound variable of the query (unfilled circle) is bounded to an individual that satisfies the restrictions imposed in the query.

3.3 Discussion

3.3.1 yOWL is an OWL-DL prototype system

yOWL is still a work in progress. In previous sections we presented the prototype of a knowledge management system in which the semantics of RDF and OWL are used to integrate and query over heterogeneous biological knowledge and we described the results obtained so far. We will now discuss in greater detail some features, lessons learned and remaining challenges.

3.3.2 Creation and population of yOWL

The first step in the design of the yOWL ontology, as described in section 2.1, was to extract, by inspection on the data obtained, the classes of entities and the relations between them. Later, the class hierarchy was augmented and refined using domain knowledge and integrated with the BFO ontology. This process could be extended towards the automated extraction of classes and relations extractions from the model available in the data source (e.g. database schema, data files, etc). Also, the creation of domain specific data wrappers is a first step towards the creation of domain independent data wrappers to populate ontologies from sources with heterogeneous data formats.

3.3.3 Gene-protein resolution

Despite SGD's recent thrust to improve its annotation of proteins [37], there is no differentiation between genes and the proteins they encode. This is problematic because an inconsistency arises when we classify genes as fiat object parts of DNA, which are disjoint with proteins as independent objects. In particular, experiments deal exclusively with genes (microarray), or proteins (two hybrid), and making strong statements as a necessary condition will lead to inconsistencies. While it will ultimately become necessary to differentiate between genes and proteins, this will require active curation or integration of knowledge which links these for yeast. In addition, since yeast biologists routinely use these two interchangeably, they might expect to see all relevant information when asking about either one, thereby requiring the formulation of more sophisticated queries.

3.3.4 Semantic knowledge integration

Given the broad scope of this ontology, we recognize that there may be subsets that overlap with other ontologies, particularly community-driven ontologies that are part of the Open Biomedical Ontologies (OBO). Unfortunately, OBO ontologies have not adopted OWL semantics including adherence to logical subsumption and we find that many are not suitable for logic-based knowledge representation and reasoning. Recent efforts through the OBO Foundry aims to redesign the OBO ontologies and map them to the BFO. Future ontology integration is already possible using OWL semantics to designate equivalent ontological classes and relations.

An ontology provides a formal conceptualization that can be used to differentiate between different types of individuals (classes). However, instances of these classes may originate from other data providers. Since OWL inherits the semantics of RDF, instances may be assigned to different namespaces. While we can state that the resource identified by UPI0000052DF0 is an instance of http://ontology.dumontierlab.com/yowl-1.0#OpenReadingFrame, we would expect that the proper namespace of that individual belongs to the original data provider, SGD in this case. One way to do this is assign a URL namespace such as http://yeastgenome.org/UPI0000052DF0. However, that specific URL does not exist, and any semantic web client would not be

able to retrieve more information about this resource. Another possibility (the one we adopted) is the use of the Life Science Identifier (LSID), a location-independent encoding of resource (URN) [38]. The advantage of this approach is that the URL resolution of the entity is done via another protocol, therefore allowing changes in URL end-points. Unfortunately, there are two issues with the LSID approach: data providers must i) register with an LSID authority directory and ii) implement a resolver that will convert the URN into a URL internet resource. Problematically, many data providers have not subscribed to the LSID resolution mechanism and therefore there may never be resolution for these entities. Compounding this problem, the LSID authority directory was not available at the time of this study, and we were forced to assign LSIDs based on the data providers root DNS. Should the data providers register with the LSID authority some future date with a different LSID, we can add another "sameAs" statement to our knowledgebase to enable data integration. Alternatively, a case might be made for new OWL semantics to make namespaces equivalent. In any case, yOWL is ready to integrate data containing LSID identifiers.

Realizing the vision of data integration requires that statements made by different sources about a single resource be considered equivalent. As outlined in the introduction, bioinformatics databases are particularly keen on maintaining their own identifiers to maintain provenance about value added contributions. This approach results in a number of equivalent identifiers for the same resources. Using the OWL property "sameAs", database cross references are made equivalent to the SGD resource. Thus, a user may query the knowledge base using any of the equivalent identifiers and return the union of statements about that resource. Some reasoners, such as Racer, provide the means to query only asserted knowledge, thereby retrieving knowledge of some subset of data providers. Such behavior is particularly well suited for users wishing to filter the knowledge base depending on the data or data provider they trust.

3.3.5 Granular semantic search

The yOWL ontology supports semantic query formulation across various levels of granularity in both class and property hierarchies. In the case of class hierarchies, a scientist may generally ask about DNA regions or query specialized DNA regions (e.g. Open Reading Frame). In the case of property hierarchies that are rooted on a non-transitive relation, one can ask whether there is any relation between two or more objects with multiple unknown concepts between them. This provides a general data mining approach to discover the shortest path between two or more resources.

3.3.6 Knowledge discovery

The ability to define classes in OWL (e.g. PathwayParticipant and PhysicalInteractionParticipant) given a set of (logically described) necessary and sufficient conditions, allows a reasoner to infer the individuals that belong to that defined class. For instance, PathwayParticipant is defined as an independent continuant that participates in some Pathway. The reasoner can also determine that a user's query corresponds to a class already defined in an ontology. Thus, individuals that belong to defined classes will be classified by the reasoner in the realization process and will not require on the fly query evaluation, which will play a role in improving query performance over greater amount of data.

3.3.7 Transitive Relations

OWL ontologies provide the ability to define transitive relations (e.g. if an Open Reading Frame is part of a Chromosome Strand, and the Chromosome Strand is part of the Chromosome, then the Open Reading Frame is part of the Chromosome). These relations are very useful in knowledge discovery. Transitive relations in relational databases are not straightforward as they require recursive SQL queries that extend relational algebra. This is hard to maintain given the information needed *a priori* (e.g. database schema, datatypes) that may limit the scope of the application, making it domain dependent. Also, the user will need to have a previous training on SQL queries, which is not very common among biology scientists.

3.3.8 Closed world and unique name assumption

Life sciences terminology often requires cardinality restrictions over properties (e.g. a carbon atom has exactly 6 protons) and negation (e.g. individuals that are DNA Regions but are not physically mapped). Moreover, life sciences ontologies are populated from databases where different names represent different entities. For these reasons, Negation as Failure (provided by RacerPro and its query language nRQL) and the ability to apply Unique Name Assumption (RacerPro and Pellet), played an important role in the query answering, especially for queries for knowledge discovery. These features are also important in the population of defined classes containing cardinality restrictions containing the "at least", "at most" and "exactly" operators.

3.3.9 Query Answering Interfaces.

The construction of semantically correct queries is facilitated by user-friendly interfaces. Protégé 4.0 offers to the users the ability to construct class queries using English phrase-like phrases (the Manchester OWL Syntax [35]). The Protégé 4.0 DL query plugin aids in the construction of the query by dynamically suggesting the phrase grammar and available entities, relations and individuals. This kind of interface helped in the construction of sophisticated queries with no training required.

Unfortunately, class queries do not return the individuals that bound the variables in the graph pattern based queries, which is essential when users want to identify multiple data that they are interested in. While Racer returns the set of individuals that satisfy the graph pattern based query, and is generally quite powerful, we found it difficult to construct the queries using the RacerPro interface and therefore we opted for the notion of graph pattern based queries to illustrate this type of query. We are aware that some efforts have been made to implement more intuitive interfaces to nRQL [23] including a graphical query language [39]. We expect this trend to continue and will facilitate the use of semantic web applications by the scientific community.

3.3.10 Scalable Data Management.

A major challenge remains with the efficient storage and retrieval of ontological data. Current applications necessarily store all data in memory to execute reasoning tasks. It will be difficult, if not impossible to store large databases with all their inferences in memory without sophisticated hardware. In our study, we necessarily restricted both the ontology size (GO slim instead of the full GO) and amount of data to reason about (5% subset). Databases such as Instance Store [40] provide a first step towards reasoning databases, but to our knowledge this is currently limited to role-free queries. More sophisticated database-driven solutions will clearly be required.

4. CONCLUSIONS

In this work, we have described a first approach to describe, integrate and query yeast biological data using the OWL-DL ontology language. To the best of our knowledge, several features make this work unique. First, we designed a domain specific ontology by extending the BFO upper level ontology and incorporate concepts from both data and expert knowledge. We maintained BFO semantics by differentiating between objects and processes, using basic relations and instantiating functions, components and processes for each gene/protein, which also helped to integrate the GO slim ontology. We also made use of RDF and OWL-DL semantics to integrate identical resources from different data providers. Finally, we illustrated the use of diverse queries at various levels of ontological granularity using OWL-DL reasoners with open and closed world semantics. This work marks a beginning for using the semantic web framework in yeast knowledge discovery. However, significant challenges remain in realizing the potential of the semantic web, such as the automated creation and population of ontologies, the efficient storage of ontological data for reasoning and the development of intuitive interfaces among others.

5. ACKNOWLEDGMENTS

This work was possible in part due to a CONACYT scholarship #150581 for Natalia Villanueva-Rosales and Carleton University startup grant for Michel Dumontier.

6. REFERENCES

1. Robu I, Robu V, Thirion B: An introduction to the Semantic Web for health sciences librarians**.** J Med Libr Assoc 2006, 94**:**198-205.
2. Berners-Lee T, Hall W, Hendler J, Shadbolt N, Weitzner DJ: Computer science. Creating a science of the Web**.** Science 2006, 313**:**769-771.
3. Berners-Lee T, Hendler J: Publishing on the semantic web**.** Nature 2001, 410**:**1023-1024.
4. Spackman KA: Normal forms for description logic expressions of clinical concepts in SNOMED RT**.** Proc AMIA Symp 2001**:**627-631.
5. Kashyap V: The UMLS Semantic Network and the Semantic Web**.** AMIA Annu Symp Proc 2003**:**351-355.
6. Bodenreider O, Mitchell JA, McCray AT: Biomedical ontologies**.** Pac Symp Biocomput 2005**:**76-78.
7. Foster I, Kesselman C, Nick J, Tuecke S: The physiology of the grid: An open grid services architecture for distributed systems integration. in Globus Project, 2002).
8. OWL-S: Semantic Markup for Web Services. http://www.w3.org/Submission/OWL-S/.
9. The Gene Ontology (GO) project in 2006**.** Nucleic Acids Res 2006, 34**:**D322-326.
10. Grenon P, Smith B, Goldberg L: Biodynamic ontology: applying BFO in the biomedical domain**.** Stud Health Technol Inform 2004, 102**:**20-38.
11. Smith B, Ceusters W, Klagges B, Kohler J, Kumar A, Lomax J, Mungall C, Neuhaus F, Rector AL, Rosse C: Relations in biomedical ontologies**.** Genome Biol 2005, 6**:**R46.
12. Wroe CJ, Stevens R, Goble CA, Ashburner M: A methodology to migrate the gene ontology to a description logic environment using DAML+OIL**.** Pac Symp Biocomput 2003**:**624-635.
13. OWL Web Ontology Language Guide. http://www.w3.org/TR/owl-guide/.
14. Horrocks I: Applications of Description Logics: State of the Art and Research Challenges. in ICCS2005 (Kassel, Germany, 2005), 78-90.
15. Sirin E, Parsia B, Grau BC, Kalyanpur A, Katz Y: Pellet: a practical owl-dl reasoner**.** in 3rd International Semantic Web Conference (ISWC2004), 2004).
16. Haarslev V, Möller R, Wessel M: Querying the Semantic Web with Racer + nRQL**.** in KI-04 Workshop on Applications on Description Logics (Ulm, 2004).
17. Wolstencroft K, Lord P, Tabernero L, Brass A, Stevens R: Protein classification using ontology classification**.** Bioinformatics 2006, 22**:**e530-538.
18. Semantic Web Best Practices and Deployment Working Group: Semantic Web Best Practices. http://www.w3.org/2001/sw/BestPractices/.
19. Alan LR: Modularisation of domain ontologies implemented in description logics and related formalisms including OWL**.** in KC2003 (Sanibel Island, FL, USA, 2003), ACM Press.
20. Aranguren ME: Ontology design patterns for the formalisation of biological ontologies. University of Manchester, Faculty of Engineering and Biological Sciences. 2005.
21. Heja G, Varga P, Pallinger P, Surjan G: Restructuring the foundational model of anatomy**.** Stud Health Technol Inform 2006, 124**:**755-760.
22. Zhang S, Bodenreider O, Golbreich C: Experience in reasoning with the foundational model of anatomy in OWL DL**.** Pac Symp Biocomput 2006**:**200-211.
23. Christopher JOB, Arash S-N, Xiao S, Volker H, Greg B: Semantic web infrastructure for fungal enzyme biotechnologists**.** Web Semant 2006, 4**:**168-180.
24. Luciano JS: PAX of mind for pathway researchers**.** Drug Discov Today 2005, 10**:**937-942.
25. Romero P, Wagg J, Green ML, Kaiser D, Krummenacker M, Karp PD: Computational prediction of human metabolic pathways from the complete human genome**.** Genome Biol 2005, 6**:**R2.
26. Joshi-Tope G, Gillespie M, Vastrik I, D'Eustachio P, Schmidt E, de Bono B, Jassal B, Gopinath GR, Wu GR, Matthews L, et al: Reactome: a knowledgebase of biological pathways**.** Nucleic Acids Res 2005, 33**:**D428-432.
27. Kulikova T, Akhtar R, Aldebert P, Althorpe N, Andersson M, Baldwin A, Bates K, Bhattacharyya S, Bower L, Browne P, et al: EMBL Nucleotide Sequence Database in 2006**.** Nucleic Acids Res 2007, 35**:**D16-20.
28. Sugawara H, Abe T, Gojobori T, Tateno Y: DDBJ working on evaluation and classification of bacterial genes in INSDC**.** Nucleic Acids Res 2007, 35**:**D13-15.
29. Wheeler DL, Barrett T, Benson DA, Bryant SH, Canese K, Chetvernin V, Church DM, DiCuccio M, Edgar R, Federhen S, et al: Database resources of the National Center for Biotechnology Information**.** Nucleic Acids Res 2007, 35**:**D5-12.
30. Babnigg G, Giometti CS: A database of unique protein sequence identifiers for proteome studies**.** Proteomics 2006, 6**:**4514-4522.
31. The Universal Protein Resource (UniProt)**.** Nucleic Acids Res 2007, 35**:**D193-197.
32. LinkHub. http://hub.gerteinlab.org.

33. Cheung KH, Yip KY, Smith A, Deknikker R, Masiar A, Gerstein M: YeastHub: a semantic web use case for integrating data in the life sciences domain**.** Bioinformatics 2005, 21 Suppl 1**:**i85-96.
34. Dwight SS, Harris MA, Dolinski K, Ball CA, Binkley G, Christie KR, Fisk DG, Issel-Tarver L, Schroeder M, Sherlock G, et al: Saccharomyces Genome Database (SGD) provides secondary gene annotation using the Gene Ontology (GO)**.** Nucleic Acids Res 2002, 30**:**69-72.
35. Horridge M, Drummond N, Goodwin J, Rector A, Stevens R, Wang HH: The Manchester OWL Syntax**.** in OWLED 2006 (Athens, Georgia, 2006).
36. Horrocks I, Sattler U, Tobies S: Reasoning with individuals for the description logic shiq**.** in 17th International Conference on Automated Deduction (CADE-17) (Germany, 2000), Springer Verlag, 482–496.
37. Nash R, Weng S, Hitz B, Balakrishnan R, Christie KR, Costanzo MC, Dwight SS, Engel SR, Fisk DG, Hirschman JE, et al: Expanded protein information at SGD: new pages and proteome browser**.** Nucleic Acids Res 2007, 35**:**D468-471.
38. Clark T, Martin S, Liefeld T: Globally distributed object identification for biological knowledgebases**.** Brief Bioinform 2004, 5**:**59-70.
39. Fadhil A, Haarslev V: GLOO: A Graphical Query Language for OWL Ontologies**.** in OWLED 2006, 2006).
40. Horrocks I, Li L, Turi D, Bechhofer S: The Instance Store: DL Reasoning with Large Numbers of Individuals. in DL2004 (Whistler, British Columbia, Canada,, 2004), CEUR-WS.org.