Journal of Biomedical Informatics 44 (2011) 1020–1031

Contents lists available at SciVerse ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Semantic integration of information about orthologs and diseases: The OGO system

Jose Antonio Miñarro-Gimenez [a], Mikel Egaña Aranguren [a,c], Rodrigo Martínez Béjar [a], Jesualdo Tomás Fernández-Breis [a,*], Marisa Madrid [b]

[a] Faculty of Computer Science, University of Murcia, Spain
[b] Paterson Institute for Cancer Research, Manchester, UK
[c] Ontology Engineering Group, Department of Artificial Intelligence, School of Computer Science, Technical University of Madrid (UPM), Spain

ARTICLE INFO

Article history:
Received 30 November 2010
Accepted 4 August 2011
Available online 16 August 2011

Keywords:
Semantic web
Bioinformatics
Biomedical ontologies
Data and knowledge integration

ABSTRACT

Semantic Web technologies like RDF and OWL are currently applied in life sciences to improve knowledge management by integrating disparate information. Many of the systems that perform such task, however, only offer a SPARQL query interface, which is difficult to use for life scientists. We present the OGO system, which consists of a knowledge base that integrates information of orthologous sequences and genetic diseases, providing an easy to use ontology-constrain driven query interface. Such interface allows the users to define SPARQL queries through a graphical process, therefore not requiring SPARQL expertise.

1. Introduction

Life sciences is a knowledge based discipline, in which the production of knowledge from data (*e.g.* the cellular location of a concrete gene product) is a daily activity. However, such knowledge is represented through vast amounts of complex and changing information stored in disparate resources [1] and in machine-unfriendly formats like natural language annotations or scientific literature [2]. The efficient management of such knowledge is paramount for the progress of research in life sciences [3], and this goal requires such knowledge to be integrated *for* humans and not *by* humans. Life scientists waste a lot of time trying to find the inter-related information, and manually removing the redundancy of the results obtained from querying different, independent information resources. In addition to this, the results provided by most of the currently available repositories are complete records rather than the concrete information units of interest for the users.

That is the situation for orthologous sequences. Orthologous sequences, or simply orthologs, are different copies of the same genetic sequence, present in different species that appeared as a result of evolutionary divergence. Nowadays, each orthology repository provides clusters of orthologous genes, which are obtained in different ways and for which several details are provided. Thus, having access to the information from those resources in an integrated way would be very useful for life scientists, as demonstrated by our previous work [4].

Orthology information is mainly used in phylogenetic studies and for taxonomic classification [5]. It is also valuable to generate new research hypotheses; *i.e.* it is likely that an ortholog of a given sequence, being in another taxon, has the same function of that sequence [6]. Therefore, orthologs are used in research about genetic diseases, in particular for understanding their genetic causes [7]. However, resources that produce and store orthologous clusters, that is, groups of related orthologs, like OrthoMCL,[1] do not offer links to genetic diseases and, consequently, the user needs to manually integrate those two kinds of information by using different tools and interfaces.

As a result, biomedical researchers need to perform a series of mainly manual tasks to retrieve this kind of translational information. For instance, retrieving the genes related to those that cause `Prostate cancer` requires performing the following actions: (1) query the Online Mendelian Inheritance in Man resource (OMIM[2]) for obtaining information about the genes that cause `Prostate cancer`; (2) query the existing orthology information resources, such as KOG [8], Inparanoid [9], OrthoMCL [10] and Homologene[3] for obtaining the orthologous genes; and (3) manual combination and analysis of the information retrieved from all orthology resources. The use of resources such as YOGY [11], which are able to submit the user query to a series of resources, present an improvement for the second step, but

* Corresponding author. Fax: +34 868884151.
E-mail addresses: jose.minyarro@um.es (J.A. Miñarro-Gimenez), megana@fi.upm.es (M. Egaña Aranguren), rodrigo@um.es (R. Martínez Béjar), jfernand@um.es (J.T. Fernández-Breis), mmadrid@picr.man.ac.uk (M. Madrid).

[1] http://orthomcl.org.
[2] http://www.ncbi.nlm.nih.gov/omim/.
[3] http://www.ncbi.nlm.nih.gov/homologene.

1532-0464/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2011.08.002

the results are not shown in an integrated way. Consequently, biomedical researchers cannot easily find this information because they are required to know: (1) which resources are available and contain the desired information; (2) how such resources can be accessed and queried; (3) the meaning of the data types and fields used in each resource. In our case, when searching for orthologs some resources are protein centered, like KOG, whereas others are gene centered, like Homologene, so this diverse of granularity may hinder the search by users. An incorrect interpretation at this level may invalidate any further analysis performed by the researcher. Thus, there is a clear need for methods and tools that help researchers in finding the right information.

One of the most promising endeavours for dealing with such knowledge management problem is the Semantic Web.[4] The Semantic Web is a vision for the next generation web, driven by the World Wide Web Consortium (W3C[5]), that advocates for a web made of data, rather than documents. The application of the Semantic Web on life sciences is steadily growing, through the so called Life Sciences Semantic Web (LSSW) [12–15,3]. The LSSW is based on the use of Semantic Web oriented W3C standards like RDF[6] (Resource Description Framework), SPARQL[7] (SPARQL Query Language for RDF) and OWL[8] (Web Ontology Language) to codify biological knowledge in a distributed and machine-friendly fashion, in Knowledge Bases (KBs) and bio-ontologies (ontologies that represent biological knowledge). An ontology is a computational representation, using a logical formalism, of a domain of discourse: it provides a collection of formally defined concepts and their relationships that can be used to integrate information or for constraints-based data validation with the support of reasoners like Pellet [16] or FaCT++ [17]. A substantial (and growing) group of current KBs and bio-ontologies exploits the LSSW approach (See [3] for a review).

In this paper, we present the new version of our Ontological Gene Orthology (OGO) system [4]. In OGO, Semantic Web technologies assist life scientists in the exploration of ortholog/genetic diseases research paths by providing a precise, explicit meaning for information units and intertwining such information. The first version of the OGO system exploited Semantic Web technologies to integrate and manage orthologs information from different resources. The work presented in this paper has improved the OGO system in the following ways. First, the semantic infrastructure has gone through major changes: (1) addition of knowledge about genetic diseases; (2) reuse of external bio-ontologies. The fact of reusing the knowledge from existing and standardized bio-ontologies makes OGO better and more interoperable. In addition to this, the classes that we are reusing from such bio-ontologies have a more precise definition and have a series of properties that were not covered in our original ontology, so the semantic infrastructure has also more knowledge now. Second, we have developed an ontology-guided, flexible query interface [18], which permits the design of queries that exploit the semantics of the OGO ontology. Therefore, researchers do not need to be aware any more of the internal structure and meaning of the data types coming from different resources, since this task is provided by the OGO system.

The overall structure and the methodology followed for building the OGO system are briefly reviewed in the beginning of Section 2. Section 3 presents the repository built in this work and also the web interfaces developed for querying the OGO Knowledge Base (OGO KB). Section 3.1 describes the detailed structure and semantics of the OGO KB. Finally, in Section 4 we discuss the issues mentioned through the paper and provide conclusions including future paths for extending the OGO system.

2. Methods

The OGO system consists of three elements: (1) an ontology about orthologs and genetic diseases (the OGO ontology), (2) a KB that stores information of orthologs and related genetic diseases using such ontology as a schema (the OGO KB), and (3) a web interface to access the system.[9] The OGO ontology is used as a scaffold to integrate information from different resources in the KB. The information is collected from orthologs databases (KOG, Inparanoid, OrthoMCL and Homologene) and the human genetic diseases database (OMIM). The integration of the information sources allows to relate the genes involved in a particular genetic disorder to their orthologs clusters. In order to extend the first version of the OGO system with the genetic disorder resource, we applied the following methodology:

1. Analysis and conceptualization of OMIM (Section 2.1).
2. Extension of the OGO ontology by including OMIM knowledge (Section 2.2).
3. Standardization of the OGO ontology by reusing external bio-ontologies (Section 2.3).
4. Definition of the mapping rules between OMIM data and the extended OGO ontology for making the data integration an automatic process (Section 2.4).

2.1. Analysis and conceptualization of OMIM

The OMIM repository consists of several files (See Table 1): `genemap`, `morbidmap`, `genetable` and `pubmed_cited`. The `genemap` file describes the cytogenetic locations of genes; the chromosome location, the genes related to the genetic disorder, the status of the OMIM disorder record and the OMIM identifier. The `morbidmap` file contains an alphabetical list of diseases in which each element in the list identifies a disease and references to other related OMIM records (linked genetic disease information can be extracted from this file). The `genetable` file relates gene names and synonyms to the OMIM record identifiers. Finally, the `pubmed_cited` file contains a list of PubMed citations related to OMIM records. The result of this analysis is the conceptualization shown in Fig. 1.

2.2. Extension of the OGO ontology with OMIM knowledge

The previous conceptualization was then compared with the OGO ontology, in order to extend it with the knowledge needed for linking the genetic disease information from OMIM to the biological information of the first version of the OGO KB. This process was manually done given the size of the OMIM conceptualization. The results of this step are described next.

Disorders have been included into the new OGO ontology, since they must be linked to already existing orthologs clusters (Fig. 1). In the new OGO ontology, two main parts can be distinguished: genetic disorders and orthologs clusters.

The main concept in this section is `Disorder`, which represents human genetic disorders from OMIM and is defined through the following properties:

- Name: the name of the disorder, which is unique and compulsory.

[4] http://www.w3.org/2001/sw/.
[5] http://www.w3.org/.
[6] http://www.w3.org/TR/rdf-primer/.
[7] http://www.w3.org/TR/rdf-sparql-query/.
[8] http://www.w3.org/TR/owl2-overview/.
[9] http://miuras.inf.um.es/ogo.

Table 1
Overall description of the data files of OMIM.

File name	Description	Version	Example

`genemap`	Cytogenetic locations of genes	28/07/09	`16.371|11|25|08|16q22.3-q23.1|ZFHX3,ATBF1|`
			`C|Zinc finger homeobox 3||`
			`104155|A, D||| Prostate cancer,susceptibility to,`
			`176807 (3)|||8(Atbf1)|`
			`16q22.3-q23.1`: Chromosomes location
			`ZFHX3, ATBF1`: Genes related to the disorder
			`C`: Status (confirmed)
			`104155`: OMIM identifier

`genemap.key`	Fields and methods used in `genemap`	19/05/09	

`morbidmap`	An alphabetical list of diseases	19/05/09	`Prostate cancer, susceptibility to, 176807 (3)|`
			`ZFHX3,ATBF1|104155|16q22.3-q23.1`
			`176807`: Reference to other related OMIM record
			`ZFHX3, ATBF1`: Genes related to the disorder
			`104155`: OMIM identifier
			`16q22.3-q23.1`: Chromosomes location

`genetable`	An alphabetical list of genes relate records	29/05/09	`tbf1 104155` `tbf1`: Gene name `104155`: OMIM identifier
`pubmed_cited`	List of PubMed articles related to records	29/05/09	`104155 1719379` `104155`: OMIM identifier `1719379`: PubMed article reference

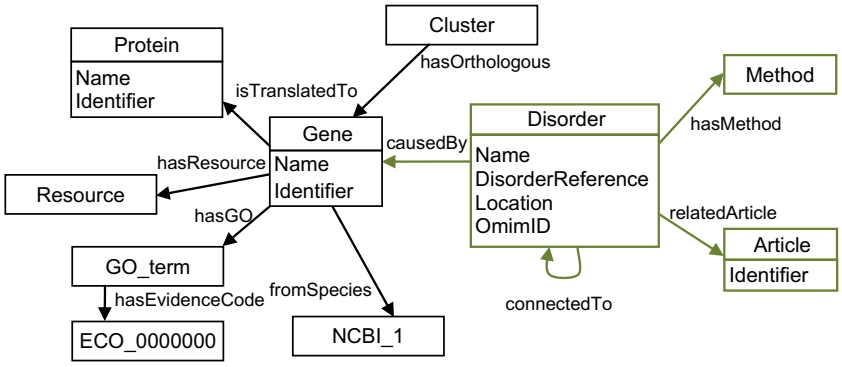

Fig. 1. The extended OGO ontology.

- OmimID: unique identifier of the record in OMIM repository, which is unique and compulsory.
- DisorderReference: links to related disorders, if any.
- Location: chromosomal location of the disorder, which is unique and compulsory.

The `Disorder` class is also connected to other classes in the ontology through the following relations (Fig. 1):

- causedBy: the gene that causes the genetic disorder, which is compulsory.
- connectedTo: connections between genetic disorders, if any.
- hasMethod: how the genes were associated with the genetic diseases (*E.g.* Deductions from the amino acid sequence of proteins). Genetic diseases are connected to at least one method through this relation.
- relatedArticle: references to research papers related to a human genetic disorder. The `Article` concept contains the identifier which is used by OMIM for referencing such article. The articles are the ones included in OMIM as Pubmed citations.

The clusters of orthologs are individuals of the class `Cluster`. The orthologs themselves are added as individuals of the class `Gene` and connected to the cluster of orthologs that they belong to through the relationship `hasOrthologous` (Fig. 1).

2.3. Standardization of the OGO ontology

The OGO ontology imports[10] other life sciences resources to inter-relate and reuse the knowledge from those resources, in order to increase the standardization of the knowledge contained in the system and to facilitate the data and knowledge interoperability.[11] For this purpose, the OWL version of some OBO ontologies have been reused. The natural translation of ontologies in OBO format[12] (GO, ECO, NCBI) into OWL treats OBO format terms like classes, not like individuals [19–21]. However, the OGO system stores entities such as genes and proteins as individuals. This means that we need to be able to refer to these entities as classes or individuals depending on the context of use. To this end, OWL punning (the ability to give the same URI to different entities[13]) was used in GO, ECO and NCBI. Therefore, by adding an individual with the same URI to every class of GO, ECO and NCBI, we could refer to such classes as values (*i.e.* as OWL individuals) or as proper OWL

[10] http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/#Imports.
[11] The OGO ontology definitions, stored in the OWL file `OGO.owl`, can be downloaded from the ontology library (http://miuras.inf.um.es/ontologies/), including the imported ones: ECO (`eco_punned.owl`), GO (`go_punned.owl`), NCBI (`ncbi_punned.owl`) and RO (`ro.owl`).
[12] http://www.geneontology.org/GO.format.shtml.
[13] http://www.w3.org/TR/2009/REC-owl2-new-features-20091027/#Simplemetamodellingcapabilities.

classes. For example, in Gene Ontology Annotation (GOA) associations like `CYC4 participates_in cell_cycle`, both `CYC4` and `cell_cycle` are OWL individuals. However, `cell_cycle` is also an OWL class.

2.3.1. Reused (Imported) ontologies

Gene Ontology (GO): GO [22] consists of three ontologies that cover the cellular component, molecular function and biological process of gene products. Orthologs are associated with GO terms via GOA associations [23].

Evidence Codes Ontology (ECO): The GOA associations are qualified with evidence codes (*e.g.* `Inferred from Physical Interaction`,`Inferred from Genomic Context`, *etc.*). ECO[14] provides a hierarchy of evidence codes, where `ECO_0000000` is the root concept, allowing querying of GOA associations *via* their evidence code types (*e.g.* a scientist may be only interested in GOA associations inferred from genomic context). Thus, the previous `Evidence_Code` concept in the OGO ontology was replaced by `ECO_0000000`. However, the `hasEvidenceCode` relationship was not changed.

NCBI taxonomy database (NCBI): The NCBI taxonomy [24] represents the taxonomical classification of organisms. In order to include the taxonomical hierarchies leading to the organisms of interest, portions of the database had to be converted to OWL. The ontological representation of biological species, and specially their classification in taxonomical ranks (genus, order, *etc.*) is still an open problem [25], due to the metamodelling nature of the biological taxonomical classification. In the case of OGO, we opted for the simplest solution, that is, codifying each taxon as an OWL class and creating a subsumption hierarchy (*e.g. Tetrapoda* is a subclass of *Sarcopterygii* in the OGO ontology), as this strategy was the least complex axiomatically. However, it should be noted that it is not a completely rigorous ontological representation.

Relationship Ontology (RO): RO [26] was created in order to provide a common set of relationships for bio-ontologies, to facilitate integration of the knowledge present in them (*e.g.* the `is-a` relationship is semantically equivalent in GO and in the Cell Type Ontology (CL) [27]). The OGO ontology imports two RO relationships, namely `participates_in` and `located_in` (`gene_product participates_in some (molecular_function or biological_process)` and `gene_product located in some cellular_component`). The rest of the needed relationships were added to the system manually, as they were not available in RO. However, relationships similar or related to the ones tailored for OGO by us have already been proposed in the RO community, even though they are still not present in the stable version of RO.[15]

GO belongs to the OBO foundry [28], *i.e.* it is a reference bio-ontology that fulfils the OBO foundry criteria.[16] ECO and RO are candidate OBO foundry ontologies. The NCBI taxonomy is not a bio-ontology *per se*, but it can be translated into OWL in a relatively straight way.

Other improvements performed on the first version of the OGO ontology are described as follows. The `Reference` class was removed and its information was integrated into the `Name` and `Identifier` properties of gene and protein concepts for clarity. In addition, the previous *fixProtein* relationship was replaced to *isTranslatedTo* for accuracy. These changes have improved the quality of the OGO ontology, adding new knowledge through the imported resources and making the modeling more clear.

2.3.2. Implementation

The OGO ontology was defined in OWL2 with Protégé,[17] a free, open source ontology editor. OWL2 was chosen for the OGO system due to its balance between expressive power and decidability, allowing the application of automated reasoning. The ontology has SRIQ(D) DL expressivity, and the combination of some and only constraints allow for closing the reasoning process for some properties.

The consistency of the OGO ontology was checked using Pellet. The consistency check involved testing the cardinality restriction, *e.g.* a gene is related to a single species and the consistency check involves testing cardinality restrictions, a gene is related to a single species, and restrictions of disjoint classes, *e.g.* gene and protein classes. Also, OWL was designed as part of the Semantic Web Stack,[18] allowing the reuse of external knowledge *via* URIs (in the case of the OGO system, GO, ECO, NCBI, and RO).

RDF is also part of the Semantic Web stack, and therefore an OWL (RDF/XML) ontology can be accessed with RDF tools, facilitating the use of both languages in a system like OGO. RDF offers the possibility of executing queries with SPARQL. The JENA Semantic Web Framework[19] was used for creating instances and querying the model through SPARQL queries. JENA is an open source Java framework for building Semantic Web applications that allows to handle OWL (RDF/XML) ontologies with persistent storage. The persistence was implemented in a MySQL[20] database: instances were stored as RDF triples in the database.

2.4. Mappings between the OGO ontology and OMIM

This process requires the definition of mapping rules between the OGO ontology and OMIM. Such rules are used for mapping the data contained in the OMIM files into their corresponding classes, properties and relationships in the OGO ontology. Once the OGO ontology was extended to include the knowledge about genetic disorders, the correspondences between the OMIM files and the ontological entities were defined, in order to make the integration of data an automatic process. The OMIM record number plays an important role in this task, as it can be used to navigate through the different OMIM files, and generate complete conceptual OMIM records. Each item is connected to the same genetic disorder instance by its OMIM identifier. Besides, each file provides different data fields representing the relations and properties of the genetic disorder. Fig. 2 depicts how the fields from OMIM files are mapped to the concepts of the OGO ontology. Once the mapping rules have been defined, the automated integration process can be launched. In Fig. 2, A, B, C and D correspond to the genemap, genetable, pubmed_cited and morbidmap files from which the genetic disease data were collected. Each one shows the data contained in the files. For instance, the morbidmap file consists of the genetic disease title, its OMIM identifier, the name of the gene that causes the disease, reference to related genetic disease records and the location of the mutation in the gene. Thus, the rules indicate how the data are interrelated to form an individual of the class genetic disease. For instance, the OMIMid of a genemap record is mapped onto the identifier of an individual of Disorder.

[14] http://www.obofoundry.org/cgi-bin/detail.cgi?id=evidencecode.
[15] See, for example http://www.bioontology.org/wiki/index.php/RO:Main_Page#Proposed_homologous_to_relation.
[16] http://www.obofoundry.org/crit.shtml.
[17] http://protege.stanford.edu/.
[18] http://en.wikipedia.org/wiki/Semantic_Web_Stack.
[19] http://jena.sourceforge.net/index.html.
[20] http://www.mysql.com/.

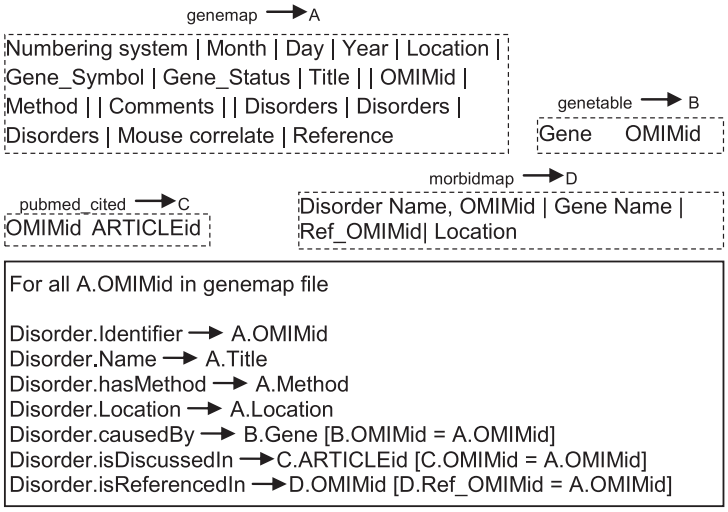

Fig. 2. Mapping rules for integrating disorder class information into the OGO KB.

3. Results

3.1. The integrated OGO system

The main page of the OGO portal[21] provides access to a brief summary of the OGO system, contact information and the query interfaces. Once we have defined the ontology model and the information has been integrated, it is possible to navigate through the classes and individuals in the ontology using the declared relationships. For example, if we have a group of orthologous genes that is linked to proteins, we can directly obtain related proteins of the orthologous group using our repository. Moreover, the query results are accurate since the information is annotated with the concepts in the ontology and we can exploit the taxonomy of biological concepts like species (NCBI), evidence codes (ECO) or the biological process, cellular component and molecular function (GO) of orthologs for defining the SPARQL query. Using SPARQL allows for defining the queries over the graph defined by the classes, properties and individuals of the ontology. Traditional systems would require the use of languages such as SQL, which would require from the query designer more knowledge about issues like how the data is distributed over the tables of the database. Besides, using such traditional repositories, we would have needed to query for the orthologous genes first and then seek for the proteins associated with those genes.

The original OGO KB contained more than 90,000 ortholog clusters, more than a million genes, and *circa* a million proteins [4]. As a result of the integration process, information about human genetic disorders was added to the OGO original KB, in the form of new instances: approximately 16,000 human genetic disorders and more than 17,000 references to PubMed citations. Therefore, in the new state of the OGO KB, for a particular gene, not only the information about its orthologous genes, but also the genetic disorders in which they are involved, can be retrieved. In the same manner, for a particular disease, not only the involved genes, but also the genes that are ortholog to them, from other organisms, can be retrieved.

An example of the queries that can be executed on the OGO repository can be seen in Fig. 3. This query searches for human genetic diseases caused by genes that are orthologs of the gene `ZFH2` and belong to the organism *Drosophila melanogaster*. The genetic diseases retrieved by the query read as follows: `Prostate cancer, susceptibility to, 104155; Ptosis, congenital, 606940; Peroxisome biogenesis factor 12, 601758`. It is possible to navigate from the relation between genetic diseases and genes to the relation of ortholog clusters in an intuitive manner. The query results were filtered by gene name and organism but they can also be filtered by GO terms, evidence codes or any other property that users may want to specify.

As it has been aforementioned, SPARQL queries search on a graph rather than on a set of tables as SQL does. Thus, we can see that there is no *from* clause in our query. In this sense, SPARQL queries are more declarative than SQL ones, because you only need to define which are the properties data must meet in order to be a match for your query, without specifying the logical path for finding the data. Both languages have *where* clause, but they are also different. In SQL, the conditions are defined at value and data type level, checking whether the column for a particular row has some value. In SPARQL, we look for triples in our graph matching the conditions, which not only are defined at data level but also at knowledge level since *ogo:causedBy* or *ogo:hasOrthologous* are semantic properties which related individuals from a class instead of concrete values.

The ontological infrastructure implemented in the OGO system would allow for a more powerful, ontology-driven semantic user interface, but semantic interfaces are not found very friendly by the average life scientist. As a matter of fact, life scientists are familiar with biological databases whose interfaces are based on keywords. Hence, we decided to develop two different types of query interfaces: one based on keywords (Section 3.2), and one driven by the ontology (Section 3.3). Both types of interfaces are described next.

3.2. Keyword-based query interface

The keyword-based query interface of OGO is shown in Fig. 4. This interface provides a text field to the users, so they can input the gene or disease of interest, and apply a series of filters. Such filters permit to limit the types of information to be part of the output (*e.g.* include/exclude the GO Terms associated to the genes), and to adjust the query (*e.g.* the genes of a particular organism). However, this interface makes a very limited use of the ontological infrastructure, since the user query is transformed into a SPARQL query using a pre-determined pattern. Given the translational nature of this repository, two versions of this basic interface have been developed, depending on the starting point for the navigation process: one for those researchers who are interested in genes

[21] http://miuras.inf.um.es/~ogo.

```
PREFIX ogo:<http://miuras.inf.um.es/ontologies/OGO.owl#>
PREFIX ncbi:<http://um.es/ncbi.owl#>
       SELECT ?disease
       WHERE
       {
            ?disease    <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>    ogo:Disorder .
            ?disease    ogo:causedBy                                         ?gene .
            ?gene       <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>    ogo:Gene .
            ?cluster    ogo:hasOrtologous                                    ?gene .
            ?cluster    ogo:hasOrtologous                                    ?ortgene .
            ?ortgene    <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>    ogo:Gene .
            ?ortgene    <http://www.w3.org/2000/01/rdf-schema#label>         "ZFH2" .
            ?ortgene    ogo:fromSpecies                                      ncbi:NCBI_7227 .
       }
```

Fig. 3. SPARQL query on OGO repository.

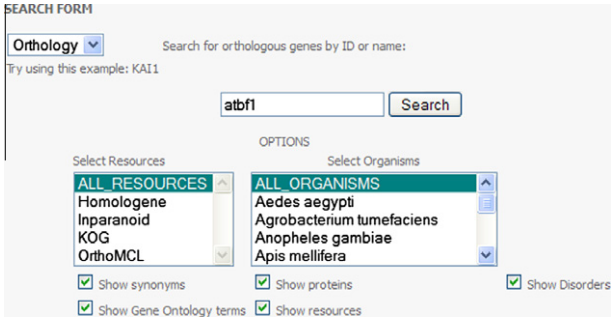

Fig. 4. Web interface for searching orthologous clusters information: the gene `ATBF1`.

(Section 3.2.1), and one for those interested in diseases (Section 3.2.2). Both interfaces constitute two different entry points to the OGO knowledge base.

3.2.1. Gene-driven query

This interface was first described in [4]. It retrieves information of orthologous clusters and it is activated by selecting `orthology` in the combo box shown on top-left of the Fig. 4, and its input is a gene name or an identifier. The retrieved information can also be filtered: organism which the gene is associated with, or the resource the gene was collected from. Moreover, some gene properties are optionally shown: alternative gene names, related proteins, the resources it can be found in, related GO terms and related genetic disorders. These attributes are retrieved by selecting the corresponding check-boxes. This interface was extended with the related genetic disorders. The interface uses a query pattern (Table 2) and the information provided by users for defining the user query.

Thus, the <*KEYWORD*> corresponds to the gene name or ID. The query filters are the <*ORGANISMn*>, which corresponds to any species in the NCBI taxonomy, and the <*RESOURCEn*>, which corresponds to any orthologs repositories. The items between square brackets ({?Disease}, {?Protein}, {?Name}, {?Resource} and {?GOterm}) are related to the check-boxes which are available in this interface.

For example, if a user wants to search for information about the gene `ATBF1`, and the associated genetic disorders, all the check-boxes should be selected in the query interface and the name of the gene should be specified in the corresponding text area. Retrieved results consist of a list of orthologous gene records which belong to the same cluster. The gene record associated with `ATBF1` is shown in Fig. 5. It contains the gene identifier, the name of its organism, the list of the other alternative gene aliases, the table with the names and identifiers of proteins, the list of information resources from which the genetic information was gathered, the list of genetic diseases, and the table of the corresponding GO terms, Evidence Codes and GOA associations that are related to it. Additional information about the items belonging to genes and genetic disorders can be accessed by clicking on its items.

Table 2
Basic orthologs query pattern.

```
@prefixogo: <http://miuras.inf.um.es/ontologies/OGO.owl>.
SELECT
?Gene0 ?Organism [?Disease ?Protein ?Name ?Resource ?GOterm]
WHERE{
?Gene1 ogo:Name ?literal.
FILTER (regex(?literal,<KEYWORD>)).
?OrthologyCluster ogo:hasOrthologous ?Gene1.
?OrthologyCluster ogo:hasOrthologous ?Gene0.
{
 ?Gene1 ogo:fromSpecies <ORGANISM1> . UNION
 ?Gene1 ogo:fromSpecies <ORGANISM2> . UNION
 ldots
 }.
 {
 ?Gene1 ogo:hasResource <RESOURCE1> .  UNION
 ?Gene1 ogo:hasResource <RESOURCE2> . UNION
 ldots
}.
?Gene0 ogo:fromSpecies ?Organism .
[OPTIONAL {?Disease ogo:causedBy ?Gene0 .}]
[OPTIONAL {?Gene0 ogo:translatedTo ?Protein .}]
[OPTIONAL {?Gene0 ogo:hasGO ?GOterm .}]
[OPTIONAL {?Gene0 ogo:Name ?Name .}]
[OPTIONAL {?Gene0 ogo:hasResource ?Resource .}]
}
```

3.2.2. Disease-driven query

This interface provides information about genetic disorders and it is activated by selecting `OMIM` in the referred combo box. In this case, the user does not input the name or identifier of a gene, but an OMIM identifier or the name of a genetic disorder. When using OMIM identifiers in queries, exact matches are retrieved. On the other hand, using the name of a genetic disorder, all the disorders that contain any input word are retrieved. For each disease included in the results, the following information is provided: name, OMIM identifier, and link to the information about the disease. For example, if the user inputs `Prostate cancer` as the query, then the result is the list of genetic disorders shown in Fig. 6.

On the other hand, if the user inputs an OMIM identifier, the information about that particular disease is shown. Thus, the interface uses the query pattern shown in Table 3 to define the query. The OMIM identifier replaces the *OMIMID* in the query. The items between brackets in the table (`?Name`, `?Location`, `?DisorderReference`, `?Gene`, `?DisorderO`, `?Method` and `?Article`) represent the properties and relationships values of a genetic disorder that are conceptualised in the KB.

Gene:
ATBF1
Organism:
Homo sapiens

Synonyms	
ID:463	
ATBT	
ZFHX3	
Protein Id	**Protein Name**
118498345	NP_008816.3
Resources	
Homologene	
Disorder	
Prostate cancer, susceptibility to [104155]	

GO Term	Evidence Code	GO Aspect
[GO:0000122] negative regulation of transcription from RNA polymerase II promoter	IGI	Biological process
[GO:0006355] regulation of transcription, DNA-dependent	TAS	Biological process
[GO:0005667] transcription factor complex	IDA	Cellular component
[GO:0005622] intracellular	IEA	Cellular component
[GO:0005634] nucleus	TAS	Cellular component
[GO:0008270] zinc ion binding	IEA	Molecular function
[GO:0043565] sequence-specific DNA binding	IEA	Molecular function
[GO:0046872] metal ion binding	IEA	Molecular function
[GO:0016564] transcription repressor activity	IGI	Molecular function
[GO:0003705] RNA polymerase II transcription factor activity, enhancer binding	TAS	Molecular function

Fig. 5. Extract of the results for the gene `ATBF1`.

Choose one of the record retrieved with "Prostate cancer" token :

- Prostate cancer, susceptibility to [104155]
- Prostate cancer, hereditary [153622]
- Prostate cancer, hereditary, 13 [157145]
- Prostate cancer 1, 176807 [180435]
- Prostate cancer, hereditary,X-linked 1 [300147]
- Prostate cancer, hereditary,X-linked 2 [300704]
- Androgen insensitivity [313700]
- Neurofibrosarcoma [600020]
- Fanconi anemia, complementation group D1 [600185]
- Prostate cancer, susceptibility to [600623]
- Prostate cancer, progression and metastasis of [600997]
- Bannayan-Riley-Ruvalcaba syndrome [601728]
- Prostate cancer, progression of [601767]
- Gastric cancer, somatic [602053]
- Lymphoma,somatic [602686]
- Prostate cancer, susceptibility to [602759]
- Li-Fraumeni syndrome [604373]
- Prostate cancer antigen 3 [604845]
- Prostate cancer, susceptibility to [605367]
- Prostate cancer aggressiveness QTL [607592]
- Prostate cancer, susceptibility to, 3 [608656]
- Prostate cancer, susceptibility to, 4 [608658]
- Prostate cancer, hereditary, 5 [609299]
- Prostate cancer, susceptibility to [609558]
- Prostate cancer, hereditary, 12 [609922]
- Prostate cancer, hereditary,7 [610321]
- Prostate cancer, hereditary,9 [610997]
- Prostate cancer, hereditary,10 [611100]
- Prostate cancer, hereditary,11 [611955]
- Prostate cancer, hereditary,14 [611958]
- Prostate cancer, hereditary,15 [611959]
- PC3 prostate cancer cell-secreted microprotein [612191]

Fig. 6. Example for Query result with the token `Prostate cancer`.

The example in Fig. 7 shows the information retrieved when querying with the OMIM identifier `104155`. The result includes the disorder name, the location of the gene in the genomic sequence, the list of alternative references to this genetic disease, the name of the gene (Other gene aliases are also provided, *e.g.* zfhx3), the list of other diseases that are associated to this disease,

Table 3
Basic genetic disorder query pattern.

```
@prefixogo: <http://miuras.inf.um.es/ontologies/OGO.owl>.
SELECT
?Disorder [?Name ?Location ?DisorderReference ?Gene ?Disorder0 ?Method
   ?Article]
WHERE{
 ?Disorder ogo:OmimID <OMIMID> .
[OPTIONAL {?Disorder ogo:Name ?Name .}]
[OPTIONAL {?Disorder ogo:Location ?Location .}]
[OPTIONAL {?Disorder ogo:DisorderReference ?DisorderReference .}]
[OPTIONAL {?Disorder ogo:causedBy ?Gene .}]
[OPTIONAL {?Disorder ogo:connectedTo ?Disorder0 .}]
[OPTIONAL {?Disorder ogo:hasMethod ?Method .}]
[OPTIONAL {?Disorder ogo:relatedArticle ?Article .}]
}
```

the list of methods used for linking genes to the disease, and the list of PubMed citations related to the disease. It is also possible to get more information about the gene, related diseases and articles by clicking on them.

3.3. The ontology-driven query interface

The ontology-driven query interface is activated by clicking on the `Advanced search` button. It allows users to include concepts, properties and relationships of the OGO ontology in the queries. This interface assists users in defining advanced SPARQL queries. This query interface consists of three sub-modules: (1) query representation module; (2) guided search module; and (3) SPARQL coding module. The query representation module is responsible for storing the query clauses defined by the users through the web interface, and for ensuring the consistency of the query since it validates the values and conditions introduced. The guided search module guides users throughout the query definition stages. It assists users in the design of the query by enabling or disabling knowledge from the query ontology. Thus, only allowed query clauses are made available to users for (re) definition of such clauses. Finally, the corresponding SPARQL query is generated and issued.

For example, the design of the query "find all orthologs that belongs to `Rattus Norvegicus` and that are also related to the gene that causes `Prostate Cancer`" is explained as follows:

1. Query concepts. In this case, we want to retrieve information about genetic diseases and their related genes. So, in order to add these concepts we have to click on the `Select Concept` button in the main interface and choose the Disorder and Gene classes from the hierarchy of classes available in the ontology to query (Fig. 8). Then, `Disorder[0]` and `Gene[1]` appear in the `Search for` text area. The numbers in brackets identify different possible instances of the same class. So, `Gene[1]` and `Gene[2]` do not have to represent the same instance of Gene class.
2. Query constraints. The filters to be applied on the knowledge base appear in the text area `Query Requirement`. The `Add requirements` button allows to add them by showing the possible properties that can be used given the context of the requirements. The context is defined by the previously defined constraints and the selected query concepts. Fig. 9 shows the allowed requirements for `Gene[1]` in the first phase of the query definition. Table 4 shows the user-defined query with the suitable requirements.
3. Query generation. Finally, the corresponding SPARQL query is generated by pressing the `Execute query` button. Table 5 shows the SPARQL query related to user-defined query in Table 4.

The results of the query (see Fig. 10) are displayed in tabular form. Each column represents the ontology classes selected in the first step of the query definition. In this case, the results are two genetic disorders and some genes that are shown grouped by the corresponding genetic disorder.

The grammar describing the query capabilities offered by the advanced interface is depicted in Table 6. The automatically generated SPARQL queries are optimized in order to reduce the response time: we sort the different types of condition clauses to make the SPARQL queries run faster. The sorting takes into account the details of our KB (it contains few concepts and relationships compared to the number of its instances). Thus, conditions which contain fewer variables and better fix a single

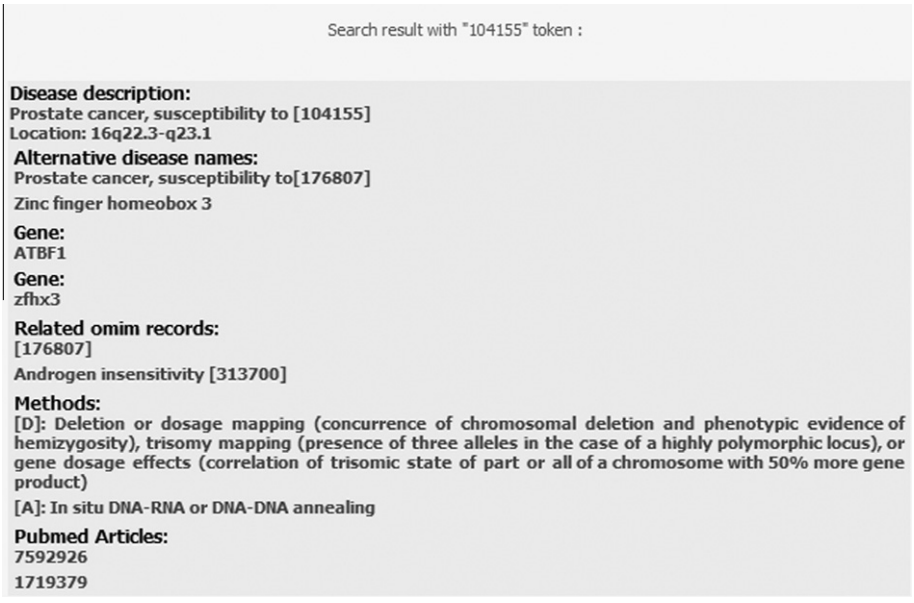

Fig. 7. Query example of a genetic disorder with the identifier `104155`.

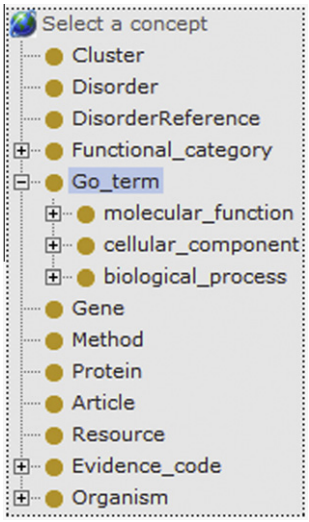

Fig. 8. Hierarchy of classes which are made available for defining an advanced query.

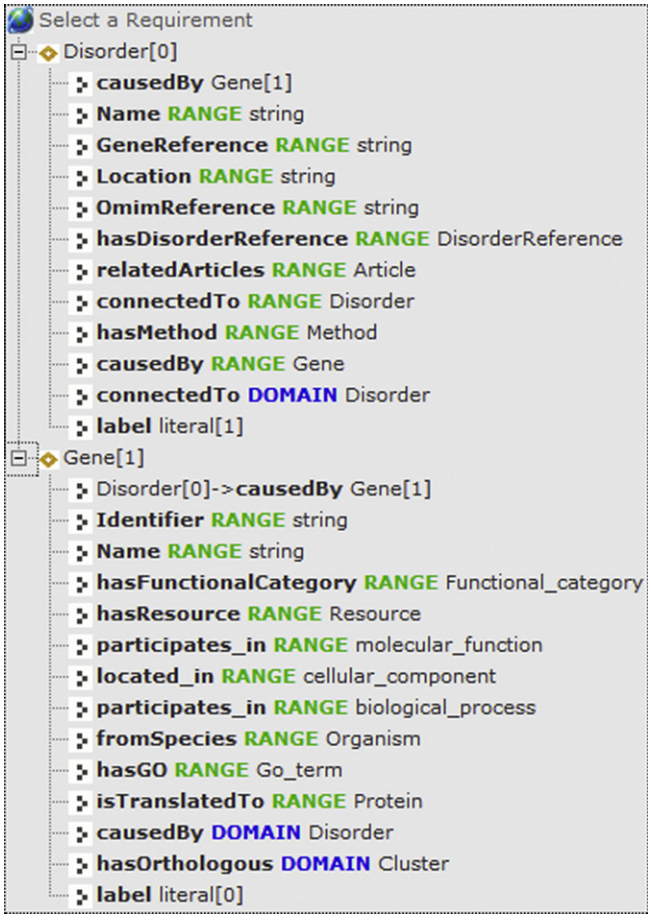

Fig. 9. Example of available requirements that can be chosen.

individual are planned to be executed first and during the sorting phase.

4. Discussion and conclusions

In this work, medical data (human genetic diseases) and biological data (orthologs clusters) were integrated and semantically

Table 4
Example of user-defined guided query.

Search for
Disorder[0]
Gene[0]
Query requirements
Disorder[0] → Name → "Prostate cancer"
Disorder[0] → causedBy → Gene[2]
Cluster[3] → hasOrthologous → Gene[2]
Cluster[3] → hasOrthologous → Gene[1]
Gene[1] → fromSpecies → "Rattus Norvegicus"

Table 5
Example of SPARQL query corresponding to Table 4.

```
@prefixogo: <http://miuras.inf.um.es/ontologies/OGO.owl>.
@prefixncbi: <http://miuras.inf.um.es/ontologies/OGO.owl>.
SELECT
  ?Disorder[0]
  ?Gene[1]
WHERE
  ?Gene[1] ogo:fromSpecies ncbi:NCBI_10116.
  ?Disorder[0] ogo:Name ?literal[4].
    FILTER(regex (?literal[4], "Prostate cancer")).
  ?Disorder[0] ogo:causedBy ?Gene[2].
  ?Cluster[3] ogo:hasOrthologous ?Gene[2].
  ?Cluster[3] ogo:hasOrthologous ?Gene[1].
}
```

represented in the OGO system. The starting point for this work was our previous research, resulting in an integrated semantic KB about orthology, the original OGO system. We reused the methodological approach described in [4] to process the extant knowledge about genetic diseases, and add it to the original KB, by improving and extending the original OGO ontology. Such extension was performed by adding new concepts and relationships about the domain of human genetic disorders to the OGO ontology. Given that both domains (human genetic disorders and orthologs clusters) are independent, the integration was free of inconsistencies and only the definition and execution of the mapping rules was necessary. The consistency of the integrated information was tested by defining domain restrictions and performing automated reasoning in order to verify its satisfaction. Therefore, this result contributes to consolidate our methodological approach, and the addition of new resources into OGO would require the execution of the same steps.

Data and knowledge integration processes, like the one described in this work, are not free of obstacles [29], like preserving consistency and avoiding redundancy. Consistency can be further divided in data consistency and knowledge consistency. In terms of data consistency, the information from OMIM and the information from the orthology resources belong to different domains. The genes are the concepts that bridge both domains, so there was limited overlap. On the other hand, knowledge consistency was guaranteed by validating the domain axioms added to the OGO ontology (*e.g.* a gene can only belong to one species). This means that our integration process is free of inconsistencies. Redundancy appears when two or more resources have defined the same gene or gene product. This fact has no impact if such duplication is detected, because genes are the basic concept in this integration process; they are the hubs that connect all the extant knowledge. Therefore, in this work, we used a strategy that has been previously followed by projects like the Cell Cycle Ontology (CCO) [6], *i.e.* to use gene products as the central item for knowledge integration.

The OGO system has been built using the Semantic Web standards RDF, SPARQL and OWL. Specifically, by codifying the information in OWL, automated reasoning was exploited as a mechanism for the automatic validation of the information. Therefore, the

Disorder[0]	Gene[1]
104155, prostate cancer, susceptibility to, zinc finger homeobox 3	pex12, 116718
	rgd1560268, zfhx3.predicted, 307829
	rgd1563022, zfhx4.predicted, 310250
600020, neurofibrosarcoma, prostate cancer, susceptibility to, max-interacting protein 1	clec5a, 679787
	ensrnog00000026306, loc684510
	loc689617, 689617
	loc689629, 689629
	max, mgc124611, 60661
	mxi.wr, mxi1, 25701
	tgap1, 294892

Fig. 10. Result of the advanced query example of Table 4.

Table 6
The grammar of the advanced query subsystem.

Query ::= "SELECT" **ListVar (WhereClause)**?
ListVar ::= Var (Var) *
WhereClause ::= "WHERE" {ConditionClause (ConditionClause) *}
ConditionClause ::= [VarCondition | **LiteralCondition]** "."
VarCondition ::= [Var | **Individual] Property [Var** | **Individual]**
LiteralCondition ::= [Var | **Individual] Property Var** "."
FILTER ("regex" **(Var "," Literal))**
Var ::=This term represents a variable in the query which can be matched to any resource of the ontology.
Individual ::=This term represents a concept or individual identified by an URI in the ontology.
Property ::=This term represents a relationship or property identified by an URI in the ontology.
Literal ::=This term represents a literal value such as STRING, INTEGER, ...

OGO integration process can facilitate consistency, *e.g.* detecting a genetic disorder which is not related to any gene; and non-redundancy, *e.g.* detecting whether a gene has already been integrated in the KB with the same name and belonging to the same species. Hence, OWL allows the integration of the additional information and resources like ECO, GO, NCBI and RO. By replacing our defined ontology concepts and relationships with those from the bio-ontologies mentioned above, the interconnection of further disparate knowledge is facilitated. Sharing a common vocabulary that prevents definitions of ambiguous terms and reusing consolidated bio-ontologies helps life scientists to understand the knowledge and to use it properly.

It should be noted that the OGO ontology does not intend to be a reference ontology for the orthology domain, to the provide the conceptualization for the domain covered by the OGO system. The standardization of the OGO ontology required the usage of the punning techique, in order to incorporate OBO format bio-ontologies into the system "as is", using the standard translation from OBO to OWL, and yet be able to exploit SPARQL queries in a straight manner. As a result, other bio-ontologies, *e.g.* CL and its cross-products against GO,[22] can be integrated in the system, also "as is", hence we would be able to exploit even more knowledge.

In the introduction section, we described the actions a researcher had to perform in order to retrieve the genes related to those that cause `Prostate cancer`. Using OGO, the researcher would only need to input `Prostate cancer` in order to receive the gene that causes the disease and, by clicking on the gene, its orthologs would be retrieved. Therefore, the OGO system is facilitating the work of researchers since the integration and comparison of information regarding orthologs and genetic diseases can be carried out automatically. By proceeding in this way, the researchers do not need to know anything about the structure and embedded meaning of the biomedical resources, since the OGO system makes it transparent.

RDF and SPARQL provide an efficient mechanism for querying semantic information, but users should not be required to learn semantic query languages for exploiting semantic repositories. This is a limitation of some systems such as BioGateway [30], the Cancer genome atlas [31] or the OpenFlyData system [32]. In fact, Semantic Web researchers have noted that "the casual user is typically overwhelmed by the formal logic of the Semantic Web" [30]. This is due to the fact that users, in order to use ontologies, have to be familiar with [33]: (1) the ontology syntax (*E.g.* RDF, OWL), (2) some formal query language (*E.g.* SPARQL), and (3) the structure and vocabulary of the target ontology. Consequently, alternative query methods are required.

Biomedical researchers are familiar with keyword-based query interfaces, because they follow the query pattern used in many available tools. This type of query interface is also provided by the OGO system, even though such interfaces exploit the domain semantics in a very limited way. Keyword-based interfaces offer fewer levels of freedom than user-based guided query definition.

[22] http://wiki.geneontology.org/index.php/XP:cellular_component_xp_cell.

Some semantic systems provide fully semantic user interfaces, but they are rather unpractical and not usable by non-expert users. Consequently, developing an advanced query interface that did not require biomedical researchers to master semantic languages was a challenge in this work. In this sense, our advanced interface allows users to properly exploit the semantics of the repository by allowing the inclusion of concepts, properties and relationships of the OGO ontology in the queries. Users are not required to write SPARQL queries, but to navigate over the OGO ontology, in order to exploit the semantics of the system.

In both types of interfaces provided by the OGO system (basic and advanced), the interface uses a query pattern that defines the types of queries that can be designed by the user (*e.g.*, gene name, gene identifier, disease name or disease identifier). However, the advanced interface provides the user with more freedom. Hence, the pattern is filled out with the information provided by the users in the text areas and check-boxes of the query interface and the corresponding SPARQL queries are generated. The current implemented SPARQL capabilities are a subset of the potential SPARQL expressivity. For instance, a subset of features, such as OR and NOT, has not yet been implemented in the interface. Besides, some axioms defined in the ontology, such as cardinality or disjointness, and properties such as label or comment are not allowed in the queries. These limitations simplify the query definition and thus provide users with only the elements of the ontology containing information about the domain.

To the best of our knowledge, there is not a system similar to the OGO system; the OGO system is a pioneering effort in the automatic integration of orthology and genetic diseases. However, even though the OGO system is a useful resource for exploring orthologs/disorders information in an integrated setting, the system can be technically improved in some areas.

Apart from for being used to integrate information or for constraints-based data validation, OWL-DL ontologies are suitable for performing automated reasoning tasks with the support of reasoners. DL reasoning is not practical currently, but the semantics of the system will be ready to exploit DL queries when reasoning becomes more efficient (something likely in the short term[23]), since we use punning. Thus, since the imported bio-ontologies are already part of the system, OWL queries exploiting inference should be able to be applied by simply increasing the efficiency of the system, without changing any semantic content. Also in the area of optimization, the OGO system can be extended with the Pellet Integrity Constraint Validator (Pellet ICV).[24] Pellet ICV interprets OWL ontologies by applying the Closed World Assumption, offering an efficient mechanism for using OWL as a schema language for validating information codified in RDF. Therefore, Pellet ICV offers a solution for checking the information gathered into the OGO system. Other extensions are planned for the OGO system. In terms of content, the OGO ontology can be extended with other bio-ontologies that also use RO, effectively integrating such bio-ontologies with the bio-ontologies already present in the OGO system, enriching it.

It should be noted that the reused ontologies have to be manually imported into the current version of the OGO system, then the data has to be integrated and updated in the OGO KB. We are planning to adapt OGO for the Linked Open Data (LOD) [34], which offers a method to publish data on the web and exploits the web in its current form to offer a single dataspace in which data can be integrated. Given that some of the resources used in OGO are currently available in the LOD cloud, we would be able to query those resources using SPARQL in real time, whereas now we need to capture the data from available files and integrate them into the OGO KB. This adaptation to the LOD would not only be benefitial for OGO but also for the LOD cloud since OGO would be publishing data that is not currently available in the LOD cloud.

In addition to this, we would like to develop methods for the definition of formal mappings between the biomedical resources and our ontology to facilitate the automatic integration of new resources of interest. For this purpose, the use of ontologies like BRO [35] would be very interesting.

In summary, the OGO system combines orthologs and human genetic diseases information, exploiting Semantic Web standards, to add value to those two types of information, by combining them. Therefore, the OGO system should be of interest, and save time, to any scientist interested in such intersection of knowledge domains.

[23] One of the OWL 2 profiles, OWL 2 QL (http://www.w3.org/TR/owl2-profiles/#OWL_2_QL), optimizes query answering for KBs with a high number of instances, like the OGO KB.

[24] http://clarkparsia.com/pellet/icv.

Acknowledgments

This work has been possible thanks to the Spanish Ministry for Science and Education through Grant TIN2010-21388-C02-02 and to the Regional Government of Murcia through Grant BIO-TEC 06/01-0005. Jose Antonio Miñarro is supported by the Seneca Foundation and the Employment and Training Service through Grant 07836/BPS/07. Mikel Egaña Aranguren is funded by the Marie Curie-COFUND Programme (FP7) of the European Commission. Marisa Madrid is supported by the EMBO Long-Term Fellowship ALTF 212-2008.

References

[1] Bodenreider O, Stevens R. Bio-ontologies: current trends and future directions. Brief Bioinform 2006;7:256–74.
[2] Attwood T, Kell D, McDermott P, Marsh J, Pettifer S, Thorne D. Rescuing knowledge lost in literature and data. Biochem J 2009;424:317–33.
[3] Antezana E, Kuiper M, Mironov V. Biological knowledge management: the emerging role of the Semantic Web technologies. Brief Bioinform 2009;10(4):392–407.
[4] Miñarro Gimenez J, Madrid M, Fernandez-Breis J. OGO: an ontological approach for integrating knowledge about orthology. BMC Bioinform 2009;10(Suppl 10):S13.
[5] Wu F, Mueller L, Crouzillat D, Petiard V, Tanksley S. Combining bioinformatics and phylogenetics to identify large sets of single-copy orthologous genes (COSII) for comparative, evolutionary and systematic studies: a test case in the euasterid plant clade. Genetics 2006;174(3):1407–20.
[6] Antezana E, Egaña M, Blondé W, Illarramendi A, Bilbao In, De Baets B, et al. The cell cycle ontology: an application ontology for the representation and integrated analysis of the cell cycle process. Genome Biol 2009;10(5):R58.
[7] Harvey K, Pfleger C, Hariharan I. The drosophila mst ortholog, hippo, restricts growth and cell proliferation and promotes apoptosis. Cell 2003;114:457–67.
[8] Tatusov R, Fedorova N, Jackson J, Jacobs A, Kiryutin B, Koonin E, et al. The cog database: an updated version includes eukaryotes. BMC Bioinform 2003;4:41–55.
[9] Remm M, Storm CV, Sonnhammer EL. Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. J Mole Biol 2001;314(5):1041–52.
[10] Chen F, Mackey A, Stoeckert C, Roos D. OrthoMCL-DB: querying a comprehensive multi-species collection of ortholog groups. Nucl Acids Res 2006;34(Suppl 1):D363–8.
[11] Penkett C, Morris J, Wood V, Bähler J. Yogy: a web-based, integrated database to retrieve protein orthologs and associated gene ontology terms. Nucl Acids Res 2006;34:W330–4.
[12] Goble C, Stevens R. State of the nation in data integration for bioinformatics. J Biomed Inform 2008;41(5):687–93.
[13] Belleau F, Nolin M, Tourigny N, Rigault P, Morissette J. Bio2RDF: towards a mashup to build bioinformatics knowledge systems. J Biomed Inform 2008;41:706–16.
[14] Good B, Wilkinson M. The life sciences semantic web is full of creeps. Brief Bioinform 2006;7(3):275–86.
[15] Ruttenberg A, Clark T, Bug W, Samwald M, Bodenreider O, Chen H, et al. Advancing translational research with the Semantic Web. BMC Bioinform 2007;8(Suppl 3):S2.
[16] Sirin E, Parsia B, Cuenca B, Kalyanpur A, Katz Y. Pellet: a practical OWL-DL reasoner. Web Semantics: Sci Serv Agents WWW 2007;5(2):51–3.
[17] Tsarkov D, Horrocks I. FaCT++ description logic reasoner: system description. Lect Notes Artif Intell 2006;4130:292–7.

[18] Miñarro Gimenez J, Egaña Aranguren M, Garcia-Sanchez F, Fernández-Breis J. A semantic query interface for the OGO platform. Lect Notes Comput Sci 2010;6266:128–42.
[19] Tirmizi S, Aitken S, Moreira D, Mungall C, Sequeda J, Shah N, et al. OBO and OWL: roundtrip ontology transformations. In: CEUR workshop proceedings; 2009. p. 559.
[20] Egana M, Bechhofer S, Lord P, Sattler U, Stevens R. Understanding and using the meaning of statements in a bio-ontology: recasting the gene ontology in owl. BMC Bioinform 2007;8:57–70.
[21] Golbreich C, Horridge M, Horrocks I, Motik B, Shearer R. OBO and OWL: leveraging semantic web technologies for the life sciences. Lect Notes Comput Sci 2007;4825:169–82.
[22] Consortium GO. Gene ontology: tool for the unification of biology. Nat Genet 2000;23(May):25–9.
[23] Camon E, Magrane M, Barrell D, Lee V, Dimmer E, Maslen J, et al. The gene ontology annotation (Goa) database: sharing knowledge in uniprot with gene ontology. Nucl Acids Res 2004;32:D262–6.
[24] Sayers E, Barrett T, Benson D, Bolton E, Stephen S, Canese K, et al. Database resources of the National Center for Biotechnology Information. Nucl Acids Res 2010;38(Suppl. 1):D5–D16.
[25] Schulz S, Stenzhorn H, Boeker M. The ontology of biological taxa. Bioinformatics 2008;24(13):i313–21.
[26] Smith B, Ceusters W, Klagges B, Köhler J, Kumar A, Lomax J, et al. Relations in biomedical ontologies. Genome Biol 2005;6(5):R46.
[27] Bard J, Rhee S, Ashburner M. An ontology for cell types. Genome Biol 2005;6:21–6.
[28] Smith B, Ashburner M, Rosse C, Bard J, Bug W, Ceusters W, et al. The OBO foundry: coordinated evolution of ontologies to support biomedical data integration. Nat Biotech 2007;25(11):1251–5.
[29] Fernandez-Breis J, Martínez-Béjar R. A cooperative framework for integrating ontologies. Int J Human–Comput Stud 2002;56(6):662–717.
[30] Bernstein A, Kaufmann E. Gino – a guided input natural language ontology editor. Lect Notes Comput Sci 2006;4273:144–57.
[31] Deus HF, Veiga DF, Freire PR, Weinstein JN, Mills GB, Almeida JS. Exposing the cancer genome atlas as a SPARQL endpoint. J Biomed Inform 2010;43(6):998–1008.
[32] Miles A, Zhao J, Klyne G, White-Cooper H, Shotton D. Openflydata: an exemplar data web integrating gene expression data on the fruit fly drosophila melanogaster. J Biomed Inform 2010;43(5):752–61.
[33] Wang C, Xiong M, Zhou Q, Yu Y. Panto: a portable natural language interface to ontologies. Lect Notes Comput Sci 2007;4519:473–87.
[34] Bizer C, Heath T, Berners-Lee T. Linked data – the story so far. Int J Seman Web Inform Syst (IJSWIS) 2009;5(3):1–22.
[35] Tenenbaum JD, Whetzel PL, Anderson K, Borromeo CD, Dinov ID, Gabriel D, et al. The biomedical resource ontology (BRO) to enable resource discovery in clinical and translational research. J Biomed Inform 2011;44(1):137–45.