

Leveraging logical rules for efficacious representation of large orthology datasets

Tarcisio M. de Farias^{1,2}, Hirokazu Chiba³, and Jesualdo T. Fernández-Breis⁴

¹Department of Computational Biology, University of Lausanne, Switzerland

²SIB Swiss Institute of Bioinformatics, Switzerland

`tarcisio.mendesdefarias@unil.ch`,

³Database Center for Life Science (DBCLS), ROIS, Japan

`chiba@dbcls.rois.ac.jp`,

⁴Departamento de Informática y Sistemas, Universidad de Murcia, IMIB-Arixaca, 30100 Murcia, Spain.

`jfernand@um.es`

Abstract. In the semantic web applied to life sciences, ontologies provide a basis to define concepts and to describe data in biological databases, thereby facilitate data interoperability across multiple resources. In the context of evolutionary genetics, the best corresponding genes across different species (e.g. the insulin genes in the pig and the human) are called “orthologs”. Dozens of bioinformatic resources identify and describe such orthologs. To represent the orthology content, an OWL-based orthology ontology (ORTH) was recently proposed. However, ORTH ontology lacks a basis to infer pairwise relations between genes besides more specific and accurate definitions of class restrictions, property domains and property ranges - which is hampering wider adoption by orthology resources. To address this issue, we present in this paper our common efforts to define a release candidate of a second version of ORTH ontology. By using this ontology, we propose a logical rule-based approach to infer information which is not explicitly defined in the primary data. As a benefit of our approach, for example, we can avoid the materialization of several billion triples to represent “is orthologous to” relation when considering the Orthologous Matrix (OMA) dataset.

Keywords: ORTH ontology, OWL, Horn-like rule, ortholog, paralog, orthology database

1 Introduction

The shared genes among different species are evidence of evolution from a common ancestor. For example, we share approximately 90% of our genes with mice. These related genes are called orthologs. Orthologs are genes in different species that evolved from a common ancestral gene by a speciation event. These genes are normally thought to retain the same function. The functional conservation of related genes across species explains the success of model organism-based research, which enables knowledge on human biology and medicine to be gained

from other species, such as mice, fruit fly, or yeast. In this context, knowledge of the orthologs between, say, mice and humans allows for studying biological processes in mice, and then transferring the knowledge to humans.

In the field of life sciences, ontologies have been identified as a key fundamental technology to achieve data interoperability across multiple resources and to annotate data, the Gene Ontology [3] being the most popular and successful one. The interest in ontologies in biomedicine can be illustrated by the fact that repositories such as BioPortal [12] contain at the time of writing more than six hundred biomedical ontologies, terminologies and controlled vocabularies. The community of orthology researchers has increased its interest for ontologies in the last years since the creation of the Quest for Orthologs (QfO) consortium¹. QfO pursues the standardization and interoperability of orthology resources and methods, including the development of common standards and formats for the representation of orthology information and knowledge. The 2013 QfO meeting [14] identified the potential benefits of semantic web technologies for the interoperability of orthology information. Since then, QfO researchers developed the first version of the Orthology Ontology (ORTH)², which served to demonstrate the feasibility of creating semantically interoperable orthology resources [6].

The experience with the ORTH has shown some limitations for the activities needed by the QfO community. More concretely, new orthology-related concepts need to be formalized in the ontology and some aspects of the current representation need to be improved in order to permit a more powerful, reasoning-based exploitation of orthology data. In this paper, we will justify why such changes are necessary in the ORTH and will present our common efforts to define a release candidate (RC) of a second version of the ORTH. Besides, we examine and compare the performance of two ways for executing queries that require inferencing. The main goal of this evaluation is to find the most appropriate approach to infer pairwise orthology relations without needing to materialize them, since that would increase significantly the number of triples to store in the already large orthology datasets. Therefore, the main contribution of this paper is how to efficaciously store orthology information using the Resource Description Framework (RDF). The extension and re-engineering of the ORTH ontology are only a step to achieve this goal.

The structure of the rest of the paper is described next. In Section 2, we will provide some background on orthology and on inferencing using semantic web content. Section 3 will present the changes made to the ORTH. The method for inferring pairwise orthology relations will be explained in Section 4. The experimental results of comparing the execution of inference-based queries to obtain pairwise orthology relations will be shown and discussed in Section 5. Finally, some conclusions will be put forward in Section 6.

¹ <https://questfororthologs.org/>

² <http://purl.org/net/orth>

2 Background

2.1 Basic concepts about orthology

Definition 1. *Homologs are genes related to each other by descent from a common ancestry. Homology is a more general term to define the relationship between genes separated by a speciation event (see Definition 2 for Ortholog) or the relationship between genes separated by a genetic duplication event (see Definition 3 for Paralog).*

Definition 2. *Orthologs are genes in different species that evolved from a common ancestral gene by speciation. The orthologs are normally thought to retain the same function in the course of evolution [7].*

Definition 3. *Paralogs are genes related by duplication. Unlike the general thought for orthologs (see Definition 2), paralogs are more likely to evolve new functions. Paralogs can be classified as inparalog and outparalog [7].*

Definition 4. *Xenologs are homologous genes that are neither orthologs nor paralogs according to above definitions, but appear to be orthologous in genome comparisons [7]. They occur due to horizontal gene transfer [15].*

Definition 5. *Hierarchical Orthologous Groups (HOGs) are defined as sets of genes that have descended from a single common ancestor within a taxonomic range of interest [2]. In the computer science context, the data structure to represent a HOG is a Tree.*

2.2 Inference-based exploitation of orthology content

There is little experience in the optimization of queries on large RDF orthology datasets. In [6], SPARQL queries were used for obtaining pairwise orthology relations, and those queries required the use of some properties defined in the ORTH in a transitive way. Such inferencing capability has to be provided by the triple store supporting SPARQL1.1. In previous works, such queries were executed over a series of graphs available in the same triple store. In [4], the authors use the ORTH to compose conjunctive queries over various knowledge bases (KBs) such as Microbial Genome Database (MBGD) ³ and Universal Protein Resource (UniProt) ⁴, although they did not investigate possible optimizations for executing inference-based SPARQL queries.

SPARQL query rewriting is a query optimization approach whose popularity has increased significantly in recent years, and it is especially useful when inferencing is an important component in the execution of the queries [9]. SPARQL query rewriting is based on changing the graph pattern included in the query, ensuring that the semantics of the query is preserved by using mappings between the query elements and the ontology. The rewriting can affect the subject, predicate or object of the triples of the query patterns.

³ <http://mbgd.genome.ad.jp>

⁴ <http://www.uniprot.org>

Languages such as SWRL⁵, RIF⁶ or SPIN⁷ also permit to use inferencing in data exploitation. SWRL and RIF permit the definition and the execution of Horn-like rules, and SPIN is built on top of SPARQL. However, neither SPARQL query rewriting or the other mentioned languages have been explored to the best of our knowledge as solutions for the exploitation of large orthology datasets.

3 Constructing the updated ontology

One of the main advantage of a DL-based ontology for knowledge representation is leveraging Horn-like rules to infer information which is not explicitly described in the primary data. In the context of recent genomics, leveraging inference enables us to store a large dataset in a compact form by retrieving implicit information on demand (see Section 4 for further details). However, the previously published ORTH ontology has several issues to be addressed in order to take advantage of the DL-based ontological representation:

1. The ORTH ontology is not fully compliant with OWL 2 DL due to ontologies imported.
2. There are not properties to describe pairwise relations between genes.
3. Missing definitions of property's domain and range.
4. Class restrictions need to be reviewed.
5. Missing several species in the imported taxonomy ontology.

In the following paragraphs, we present how we solve those issues. For the sake of simplicity, in the rest of this paper we omit the namespace prefixes whenever it does not compromise the understandability.

DL compliance. The first release of the ORTH ontology⁸ asserts that *rdfs:Resource* $\sqsubseteq \top$ (i.e. *rdfs:Resource* a *owl:Class*) and $\top \sqsubseteq \forall hasSource.rdfs:Resource$. Nevertheless, in the OWL 2 DL profile for the sake of decidability, an entity can not be an instance and a class at the same time. As a reminder, the *rdfs:Resource* is an instance of *rdfs:Class* and *owl:Class* is a subclass of *rdfs:Class*. Therefore, not all RDFS classes are legal OWL DL classes. Although, in terms of data modeling this issue is not a relevant problem, without fixing this we can not take advantage of the available reasoning tools. These tools are fundamentally important to our Horn-like rule-based approach presented in Section 4. To address this first issue, we removed the axioms *rdfs:Resource* $\sqsubseteq \top$ and $\top \sqsubseteq \forall hasSource.rdfs:Resource$.

Pairwise relations. In genetics, we can relate genes according to a common ancestral DNA sequence such as homolog, ortholog, paralog, xenolog, in-paralog and outparalog relationships. The first version of ORTH ontology permits to obtain the pairwise relations by means of SPARQL queries over the

⁵ <https://www.w3.org/Submission/SWRL/>

⁶ <https://www.w3.org/TR/rif-overview/>

⁷ <http://spinrdf.org/>

⁸ <https://bioportal.bioontology.org/ontologies/ORTH>

semantic, representation of the HOGs, but does not contain properties to assert these relations between genes. However, being able to represent, persist and exploit such relations is needed for some exploitation scenarios. To be able to represent the pairwise relations, we include the axioms in Listing 3.1.

$$\begin{aligned} \top &\sqsubseteq \forall hasHomolog. SequenceUnit \\ \exists hasHomolog. \top &\sqsubseteq SequenceUnit \\ hasOrtholog &\sqsubseteq hasHomolog \\ hasParalog &\sqsubseteq hasHomolog \\ hasXenolog &\sqsubseteq hasHomolog \end{aligned}$$

Listing 3.1. The axioms added to describe homologous pairwise relations.

Similar properties to *hasHomolog*, *hasOrtholog* and *hasParalog* already exist in the Semanticscience Integrated Ontology (SIO) ontology. However, SIO does not specify the domain and range of these properties. Moreover, SIO is a more general purpose ontology, it has been reused in ORTH. Nonetheless, for the sake of interoperability, we can state that the ORTH ontology pairwise relations are subproperties of their correspondent SIO properties when exist.

Property and class restrictions. To exemplify a property’s range modification, we modified the range of the *hasCluster* property from *GeneTreeNode* into *HomologsCluster* class. This is because the property value must not be a gene but a cluster. Further details of changes in class restrictions and property’s domain and range in the ORTH ontology are available on the following URL: <https://github.com/qfo/OrthologyOntology>.

Species taxonomy ontology. The NCBI organismal taxonomy ontology used in the first version of ORTH ontology refers to a view of the NCBITaxon ontology⁹. Thus, it does not describe an exhaustive list of species. Because of this, we replaced the *NCBI_1* class with *NCBITaxon_1* that is the root taxonomy class in the NCBITaxon ontology.

Several classes in life sciences related ontologies are not supposed to be instantiated or they are singleton classes (i.e. the class is only instantiated once). Some examples are the classes of the following life science ontologies: Gene Ontology [3], UBERON ontology [11], SIO ontology and also NCBITaxon ontology. Therefore, when importing the NCBITaxon ontology along with the new version of ORTH ontology, one instance must be created for each species classes to assign the ‘*in taxon*’ property for a *SequenceUnit* instance, which is done using the Punning¹⁰ feature of OWL 2. This class instantiation is necessary to be DL compliant because a NCBITaxon class can not be directly assigned to the ‘*in taxon*’ property. As a reminder, only an instance can be a value of an object property. Further analysis of the drawbacks of defining a large Terminological Box (TBox) with singleton classes instead of having a smaller TBox with a relevant Assertional Box (ABox) are beyond of the scope of this paper. For information, the NCBITaxon ontology contains about 1,600,000 classes.

To build the new RC ORTH Ontology, we made 27 modifications in the previous ORTH ontology version that include adding and removing properties, prop-

⁹ <http://www.obofoundry.org/ontology/ncbitaxon.html>

¹⁰ https://www.w3.org/TR/owl2-new-features/#F12:_Punning

erty domain, property range, classes and class restrictions. A full description of these modifications is available on <https://github.com/qfo/OrthologyOntology>. The RC ORTH ontology is available to download on the following URL: http://purl.org/net/orth_rc.

4 Inferring pairwise relations from hierarchical structures

End-users are typically interested in pairwise relationships such as “is orthologous to”. Because of this, from now on by considering the RC ORTH ontology (DL-based) that is described in Section 3, we can assert pairwise relations between genes. However, today’s orthology information providers store all pairwise relationships, which grow quadratically with the number of genes or genomes. To address this problem, we capture the implicit information of pairwise relationships with an inference engine. This information is implicitly structured in HOGs (see Section 2 for further details). In doing so, the data to be stored and retrieved scales linearly. For example, we do not need to store pairwise orthologs between species because they can be inferred by applying the R1 Horn-like rule shown in Listing 4.1. Thus, with our approach we can infer new information instead of materializing it. For example, we can avoid the materialization of 6,464,814,646 triples to explicitly define orthologous relationships when considering solely 1,048,561 out 4,172,982 orthologous clusters in the latest Orthologous Matrix (OMA) database (DB) release. For comparison reasons, by using the HOGs, we solely need 16,911,449 triples to implicitly define the pairwise orthologs from HOGs in OMA.

```

R1: OrthologsCluster(cluster) ^ hasHomologousMember(cluster, node1) ^ hasHomologousMember
    (cluster, node2) ^ 'has part'(node2, seq2) ^ 'has part'(node1, seq1) ^ SequenceUnit(seq1) ^
    SequenceUnit(seq2) ^ (node1 ≠ node2) → hasOrtholog(seq1, seq2)

R2: ParalogsCluster(cluster) ^ hasHomologousMember(cluster, node1) ^ hasHomologousMember
    (cluster, node2) ^ 'has part'(node2, seq2) ^ 'has part'(node1, seq1) ^ SequenceUnit(seq1) ^
    SequenceUnit(seq2) ^ (node1 ≠ node2) → hasParalog(seq1, seq2)

```

Listing 4.1. The Horn-like rules that infers the *hasOrtholog*(R1) and *hasParalog*(R2) properties for a given *SequenceUnit* instance (e.g. *Gene* instance).

Listing 4.2 contains the equivalent subquery to the R1 rule in Listing 4.1 to retrieve the implicit *hasOrtholog* assertions. This subquery can be used with a SPARQL query rewrite approach [8] to infer the *hasOrtholog* relations between genes (or proteins). Therefore, it is an alternative solution to a general purpose inference engine. For example, triple stores which does not fully support reasoning can consider Listing 4.2 subquery to replace the occurrences of *hasOrtholog* in the original SPARQL query. For example, let us suppose the following SPARQL query *SELECT * { ?g1 :hasOrtholog ?g2. ?g1 :geneName 'APOC1' }*. By parsing this query, a SPARQL query rewrite approach identifies the basic graph pattern (BGP) *?g1 :hasOrtholog ?g2* that is replaced with the graph between braces in Listing 4.2 by also considering variable names (e.g. *?seq_1* is replaced with *?g1*). The expanded query is then executed in a SPARQL

endpoint (i.e. triple store). Moreover, in Section 5, we present the performance in terms of query execution time and retrieved results along with a discussion about the benefits and drawbacks of both approaches.

```

SELECT ?seq_1 ?seq_2 {
  ?cluster a :OrthologsCluster.
  ?cluster :hasHomologousMember ?node_1.
  ?cluster :hasHomologousMember ?node_2.
  ?node_1 :hasHomologousMember* ?seq_1.
  ?node_2 :hasHomologousMember* ?seq_2.
  {?seq_1 a :Gene. ?seq_2 a :Gene.} UNION
  {?seq_1 a :Protein. ?seq_2 a :Protein.}
  FILTER (?node_1 != ?node_2)

```

Listing 4.2. The subquery to assert the `hasOrtholog` property for a given *SequenceUnit* instance (e.g. *Gene* or *Protein* instance).

The R2 rule in Listing 4.1 is a Horn-like rule to infer *hasParalog* property. The equivalent SPARQL subquery for `hasParalog` is similar to the subquery in Listing 4.2 except by the fact that the first triple in Listing 4.2 `?cluster a :OrthologsCluster` is replaced with `?cluster a :ParalogsCluster`.

Some resources actually use orthologous clusters as homologous clusters. To solve this issue at the query level, we can add a condition in the R1 rule in Listing 4.1 and the query in Listing 4.2 to only consider genes/proteins in different species (i.e. orthologs). Nevertheless, the concepts of homolog and ortholog should not be misleading.

As a consequence of our proposed Horn-like rule-based approach, we can also make it easier to write queries for retrieving orthology information since the second version of the ORTH ontology is a more fine-grained ontology. There are property values assigned by applying Horn-like rules (e.g. Semantic Web Rule Language rules) at query execution time.

5 Results and Discussion

To further justify the gain in terms of storage by inferring pairwise relations instead of materializing them, we inferred about 8,034,238,900 *hasParalog* assertions between proteins in the OMA DB by considering the R2 rule in Listing 4.1. These inferred assertions also consider the symmetric inferences (i.e. if *A hasParalog B* then *B hasParalog A*). Therefore, with the ORTH ontology based on HOGs, we can efficaciously represent RDF-based homology relations such as *hasParalog* and *hasOrtholog*.

The experiment has consisted on comparing the time performance of SPARQL query rewrite and DL-safe [10] Horn-like rule based approaches. For this purpose we have used the subqueries presented in Section 4. Each query has been executed thirty times for each approach. We have solely considered one OMA HOG at the LUCA taxonomic level, so containing 2,727 proteins. In this experiment, we have used the Stardog 5 triple store [1] with 6GB of dedicated RAM memory. All the tests were run in a computer with 3.5GHz dual-core Intel Core i7 processor, Turbo Boost up to 4.0GHz, 16GB of 2133MHz LPDDR3 memory

and 1TB SSD. The choice of the Stardog is due to the fact that it supports DL-safe Horn-like rules combined with OWL2 constructs and reasoning at query execution time [5, 13].

We executed the Q1 and Q2 queries in Listing 5.1 by using a SPARQL query rewrite approach and the Stardog’s DL-safe rule inference engine. The Q1 query retrieves all *hasOrtholog* relations of the protein with the *HUMAN29522* OMA identifier. This protein is the cytochrome c oxidase subunit 1 encoded by the MT-CO1 gene. Table 1 presents the results obtained in terms of query execution time in milliseconds (mean and standard deviation) and the number of retrieved results for the 30 executions of Q1 and Q2 queries. The Q2 query (see Listing 5.1) retrieves all *hasParalog* relations for the same protein (i.e. *HUMAN29522*).

```
Q1: SELECT ?seq_1 { ?seq_1 orth:hasOrtholog oma:PROTEIN_HUMAN29522 }
Q2: SELECT ?seq_1 { ?seq_1 orth:hasParalog oma:PROTEIN_HUMAN29522 }
```

Listing 5.1. Querying the orthologous (Q1) and paralogous (Q2) genes of MT-CO1 human gene in OMA database.

From Table 1, we can conclude the SPARQL query rewrite approach is ≈ 106 ms and ≈ 40 ms faster in average than the DL-safe rule based approach to retrieve the same amount of *hasOrtholog* and *hasParalog* assertions, respectively. As a reminder, for the results in these tables, we only considered the HOG that contains the *HUMAN29522* protein. Although, there are 589,223 HOGs in OMA DB. Table 2 shows the results of executing the queries in Listing 5.1 taking into account all OMA HOGs and using a timeout of 5 minutes.

Query	Approach	Mean time(ms)	Std deviation (σ)	#Results
Q1	SPARQL query rewrite	193.7	33.8	2,722
Q1	DL-safe rule based	300.3	78.1	2,722
Q2	SPARQL query rewrite	65.1	13.0	4
Q2	DL-safe rule based	104.6	17.8	4

Table 1. Performance comparison between SPARQL query rewrite and DL-safe Horn-like rule based approaches for Q1 and Q2 queries in Listing 5.1.

Table 2 demonstrates that the DL-safe Horn-like rule based approach is not able to retrieve any results after 5 minutes of query execution by using the Stardog triple store. This is mainly because the Horn-like rules to infer *hasParalog* and *hasOrtholog* relations contain a transitive property labeled as “has part” instead of the *:hasHomologousMember** SPARQL property path¹¹ (see query in Listing 4.2). The performance issues are due to the fact that Stardog processes first the ‘has part’ transitive property that does not contain any subject or object assigned. Therefore, Stardog attempts to infer all possible ‘has part’ assertions over all HOGs to afterwards apply the join operations. As a reminder, for the tests in Table 2, we are considering the whole OMA DB that contains 9,443,947 proteins without counting alternative splicing. This explains why the DL-safe rule based approach based on Stardog is not capable of retrieving any result in some milliseconds. However, by using *:hasHomologousMember** SPARQL

¹¹ <https://www.w3.org/TR/sparql11-property-paths/>

property path, Stardog calculates the query execution plan better as justified in Table 2. Because of this, Stardog’s SPARQL processor retrieves all results in milliseconds. This also justifies why the SPARQL query rewrite approach had better results than the DL-safe rule based one in Table 1 when considering only one HOG.

Query	Approach	Mean time(ms)	Std deviation (σ)	#Results
Q1	SPARQL Query rewrite	216.5	109.5	2,722
Q1	DL-safe rule based	300,000	-	-
Q2	SPARQL Query rewrite	66.8	16.2	4
Q2	DL-safe rule based	300,000	-	-

Table 2. Performance comparison between SPARQL query rewrite and DL-safe Horn-like rule based approaches for Q1 and Q2 queries in Listing 5.1 by considering the entire OMA database.

Despite the Stardog’s results depicted in this section to process transitive properties, the main benefit of using the Horn-like rule based approach described in Section 4 is the possibility of reusing inferred concepts and properties to define other Horn-like rules. This can be done in a modular way similar to a function in traditional programming languages (e.g. C language). Therefore, implicit information in an orthology database becomes explicit by defining these logical rules. Another benefit is the fact that we can take advantage of general purpose inference engines to process the Horn-like rules.

6 Conclusion

To build the RC of a second version of the ORTH ontology, we made 27 modifications in the previous ORTH version that include adding and removing properties, property domain, property range, classes and class restrictions. We also discussed how the ORTH ontology should be instantiated to avoid for example non-compliance with DL due to imported ontologies. Moreover, we described the benefits of using a rule based approach to infer new information from the orthology data. In doing so, we can drastically reduce the number of stored triples, facilitate the work of writing SPARQL queries and reuse inferred properties to define new rules. We also argue about performance issues of a Horn-like rule based approach compared to a query rewrite approach. Although our experiments by using Stardog show that a SPARQL query rewrite approach is more efficient, we cannot conclude it is significantly better than a DL-safe Horn-like rule-based one. This is because Stardog does not calculate the query execution plan in the same way as for transitive properties and SPARQL property path.

One final remark is concern about performing the tests in Section 5 by using alternative triple stores that support Horn-like rules combined with OWL 2 constructs and perform reasoning at query execution time. In future work we will consider annotating the ORTH entities by harnessing natural language processing and keyword searching techniques.

Acknowledgements

This work has been financed by the Swiss National Research Programme (NFP) 75 (see <http://www.nfp75.ch>) - SNSF Project 167149. Part of the work was supported by the ROIS International Networking project and conducted through NBDC/DBCLS BioHackathon 2017 (see <http://www.biohackathon.org>).

References

1. Complexible Inc. : Stardog 5: The manual (2017) Available online: <http://docs.stardog.com/>. Last accessed on October, 10th 2017.
2. Altenhoff, A.M., Gil, M., Gonnet, G.H., Dessimoz, C.: Inferring hierarchical orthologous groups from orthologous gene pairs. *PLoS One* **8**(1) (2013) e53786
3. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., et al.: Gene ontology: tool for the unification of biology. *Nature genetics* **25**(1) (2000) 25
4. Chiba, H., Uchiyama, I.: Spang: a sparql client supporting generation and reuse of queries for distributed rdf databases. *BMC bioinformatics* **18**(1) (2017) 93
5. de Farias, T.M., Roxin, A., Nicolle, C.: Swrl rule-selection methodology for ontology interoperability. *Data & Knowledge Engineering* **105** (2016) 53–72
6. Fernández-Breis, J.T., Chiba, H., del Carmen Legaz-García, M., Uchiyama, I.: The orthology ontology: development and applications. *Journal of biomedical semantics* **7**(1) (2016) 34
7. Koonin, E.V.: Orthologs, paralogs, and evolutionary genomics. *Annu. Rev. Genet.* **39** (2005) 309–338
8. Makris, K., Gioldasis, N., Bikakis, N., Christodoulakis, S.: Ontology mapping and sparql rewriting for querying federated rdf data sources. *On the Move to Meaningful Internet Systems, OTM 2010* (2010) 1108–1117
9. Makris, K., Gioldasis, N., Bikakis, N., Christodoulakis, S.: Sparql rewriting for query mediation over mapped ontologies. *Technical University of Crete* (2010)
10. Motik, B.: Reasoning in description logics using resolution and deductive databases. PhD thesis
11. Mungall, C.J., Torniai, C., Gkoutos, G.V., Lewis, S.E., Haendel, M.A.: Uberon, an integrative multi-species anatomy ontology. *Genome biology* **13**(1) (2012) R5
12. Noy, N.F., Shah, N.H., Whetzel, P.L., Dai, B., Dorf, M., Griffith, N., Jonquet, C., Rubin, D.L., Storey, M.A., Chute, C.G., et al.: Bioportal: ontologies and integrated data resources at the click of a mouse. *Nucleic acids research* **37**(suppl.2) (2009) W170–W173
13. Pauwels, P., de Farias, T.M., Zhang, C., Roxin, A., Beetz, J., De Roo, J., Nicolle, C.: A performance benchmark over semantic rule checking approaches in construction industry. *Advanced Engineering Informatics* **33** (2017) 68–88
14. Sonnhammer, E., Gabaldón, T., Sousa da Silva, A., Martin, M., Robinson-Rechavi, M., Boeckmann, B., Thomas, P., Dessimoz, C.: Big data and other challenges in the quest for orthologs. *Bioinformatics* **30**(21) (2014) 2993–2998
15. Soucy, S.M., Huang, J., Gogarten, J.P.: Horizontal gene transfer: building the web of life. *Nature Reviews Genetics* **16**(8) (2015) 472–482