Check for updates

METHOD ARTICLE

A hands-on introduction to querying evolutionary relationships across multiple data sources using SPARQL [version 1; peer review: 1 approved, 2 approved with reservations]

Ana Claudia Sima[1-3], Christophe Dessimoz[2-6], Kurt Stockinger[1], Monique Zahn-Zabal[2,3], Tarcisio Mendes de Farias[2-4,7]

[1]ZHAW Zurich University of Applied Sciences, Winterthur, Zurich, Switzerland
[2]Department of Computational Biology, University of Lausanne, Lausanne, Vaud, Switzerland
[3]SIB Swiss Institute of Bioinformatics, Lausanne, Vaud, Switzerland
[4]Center for Integrative Genomics, University of Lausanne, Lausanne, Vaud, Switzerland
[5]Department of Computer Science, University College London, London, UK
[6]Department of Genetics, Evolution, and Environment, University College London, London, UK
[7]Department of Ecology and Evolution, University of Lausanne, Lausanne, Vaud, Switzerland

v1 **First published:** 29 Oct 2019, **8**:1822
https://doi.org/10.12688/f1000research.21027.1

Latest published: 22 Jul 2020, **8**:1822
https://doi.org/10.12688/f1000research.21027.2

Abstract

The increasing use of Semantic Web technologies in the life sciences, in particular the use of the Resource Description Framework (RDF) and the RDF query language SPARQL, opens the path for novel integrative analyses, combining information from multiple sources. However, analyzing evolutionary data in RDF is not trivial, due to the steep learning curve required to understand both the data models adopted by different RDF data sources, as well as the SPARQL query language. In this article, we provide a hands-on introduction to querying evolutionary data across multiple sources that publish orthology information in RDF, namely: The Orthologous MAtrix (OMA), the European Bioinformatics Institute (EBI) RDF platform, the Database of Orthologous Groups (OrthoDB) and the Microbial Genome Database (MBGD). We present four protocols in increasing order of complexity. In these protocols, we demonstrate through SPARQL queries how to retrieve pairwise orthologs, homologous groups, and hierarchical orthologous groups. Finally, we show how orthology information in different sources can be compared, through the use of federated SPARQL queries.

Keywords

Orthology, Comparative Genomics, Sequence Homology, Resource Description Framework (RDF), SPARQL

This article is included in the The OMA collection collection.

Open Peer Review

Reviewer Status ? ? ✓

	Invited Reviewers		
	1	2	3
version 2 (revision) 22 Jul 2020			
version 1 29 Oct 2019	? report	? report	✓ report

1 **Andra Waagmeester**, Micelio, Antwerp, Belgium

2 **Ikuo Uchiyama**, National Institute for Basic Biology, National Institutes of Natural Sciences, Okazaki, Japan

3 **Tatsuya Kushida**, National Bioscience Database Center, Tokyo, Japan
RIKEN BioResource Research Center, Tsukuba, Japan

Any reports and responses or comments on the article can be found at the end of the article.

Corresponding author: Tarcisio Mendes de Farias (tarcisio.mendesdefarias@unil.ch)

Author roles: Sima AC: Conceptualization, Methodology, Software, Writing – Original Draft Preparation, Writing – Review & Editing; **Dessimoz C**: Funding Acquisition, Writing – Review & Editing; **Stockinger K**: Funding Acquisition, Writing – Review & Editing; **Zahn-Zabal M**: Writing – Review & Editing; **Mendes de Farias T**: Conceptualization, Formal Analysis, Investigation, Methodology, Project Administration, Software, Supervision, Validation, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This work was funded by the Swiss National Research Programme 75 “Big Data” (Grant 167149) and a Swiss National Science Foundation Professorship grant to CD (Grant 150654).
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

How to cite this article: Sima AC, Dessimoz C, Stockinger K *et al.* **A hands-on introduction to querying evolutionary relationships across multiple data sources using SPARQL [version 1; peer review: 1 approved, 2 approved with reservations]** F1000Research 2019, **8**:1822 https://doi.org/10.12688/f1000research.21027.1

First published: 29 Oct 2019, **8**:1822 https://doi.org/10.12688/f1000research.21027.1

Introduction

Gene classification based on evolutionary history is essential for many aspects of comparative and functional genomics - reviewed in (Gabaldón & Koonin, 2013); (Glover *et al.*, 2019). On the one hand, evolutionary relations are often described as binary relations. Two genes that share a common ancestor are defined as *homologs*. We can classify homologs into *orthologs*, which originate from a speciation event; *paralogs*, which originate from a gene duplication; and *xenologs*, which originate from horizontal gene transfer (Fitch, 1970). On the other hand, Hierarchical Orthologous Groups (HOGs) are hierarchical clusters of corresponding genes where each level in the hierarchy refers to a common ancestral gene at a taxonomic level of reference (Altenhoff *et al.*, 2013). Identifying orthologs and HOGs is valuable in several contexts such as gene function inference, gene evolution dynamics and comparative genomics.

To query and interoperate biological databases, Semantic Web Technologies are being increasingly adopted, in particular the use of the Resource Description Framework (RDF) and SPARQL protocol and RDF query language (SPARQL Query Language for RDF, n.d.). However, despite the progress they have enabled in several fields, particularly in the life sciences (Duek *et al.*, 2018), (Iyappan *et al.*, 2016) (Williams *et al.*, 2012), there are still significant challenges that limit their use for the larger scientific community. In particular, analysing evolutionary relationship data in RDF poses the following challenges:

1) *complex* data models - for example, while storing data in a hierarchical structure (HOGs) results in significant performance benefits for common analyses, such as computing orthologs of a specific gene in a different model organism, the hierarchy also results in requiring advanced knowledge of the SPARQL language (in particular, recursivity) in order to benefit from the RDF representation of HOGs. In this article, we present a series of hands-on examples, in increasing order of complexity, to familiarise the reader with the basic concepts needed to query evolutionary relationships in orthology databases.

2) *heterogeneous* data models - understanding the data model of a *single* orthology database might not be sufficient in general, since different databases have made different design decisions. We help overcome this challenge by depicting how the following major Orthology Databases structure their data in RDF, as well as how they can be queried using SPARQL: the Orthologous MAtrix (OMA) (Altenhoff *et al.*, 2018), the European Bioinformatics Institute (EBI) RDF platform (Jupp, *et al.*, 2014), the Database of Orthologous Groups (OrthoDB) (Zdobnov *et al.*, 2017) and the Microbial Genome Database (MBGD) (Uchiyama *et al.*, 2018).

3) *overhead of integration* into existing analysis pipelines. The limited rate of adoption of Semantic Web Technologies can be explained by the reluctance of bioinformaticians to change their existing workflows in order to accommodate new data formats based on the RDF framework. For example, retrieving orthology information using public SPARQL endpoints instead of the more traditional file-based data exchange or full database dumps. A SPARQL endpoint is an access point for receiving and processing SPARQL protocol requests. In this article, we show through concrete examples that integrating the results of SPARQL queries into existing analyses is a straightforward task - more specifically, we show how to transform the results into regular Pandas dataframes in Python. Furthermore, we provide an accompanying Jupyter notebook where all the examples presented in this article can be directly tested and further refined.

This article has several goals:

(1) *Understanding Orthology Data Models.* Become familiar with how evolutionary relationships are represented in RDF across several databases. Learn about the data modelling decisions: common points as well as differences between these sources to support the choice of one or more of them for a given analysis.

(2) *Understanding how to query orthology data using SPARQL.* To this end, we extend the introduction and examples in (Chiba *et al.*, 2015) (de Farias *et al.*, 2017), while also covering multiple, distributed orthology data sources.

(3) *Integrating external sources.* Leverage connections to other external bioinformatics resources that make their data available in public SPARQL endpoints based on cross-references. In particular, learn about the role of UniProt cross-references as a *bridge* between different data sources in integrative analyses.

In addition, we show how to use SPARQL to make meta-analyses combining multiple orthology databases. For instance, for a given gene, which are the orthologs in a given database which are *not* present in another one? Finally, we show how to leverage SPARQL aggregations in order to get useful statistics about orthology data available in the sources.

Finally, learn how to leverage SPARQL results in downstream analyses by converting them to Pandas dataframes. This is illustrated through a series of hands-on exercises in the accompanying Jupyter Notebook (exercises provided in Python).

The protocols presented in this article are aimed at bioinformaticians who are already familiar with the basics of SPARQL and wish to learn how orthology data can be integrated in their research analyses programmatically, through the use of (federated) SPARQL queries.

Materials

In the following paragraphs, we briefly describe the orthology databases considered in this article.

OrthoDB (Zdobnov *et al.*, 2017) contains orthologous genes along with evolutionary and functional annotations. It relies on HOGs to enable different orthology information resolutions with regards to more closely related species. The 2018 OrthoDB version covers thousands of eukaryotes, prokaryotes, and viruses. OrthoDB data is available in RDF through the public SPARQL endpoint at https://sparql.orthodb.org/sparql. We note here that the timeout for the public SPARQL endpoint is limited to 100 seconds - more precisely, queries with longer *estimated* execution time will *not* be allowed to run.

MBGD (Uchiyama *et al.*, 2018) is a comparative genomics database that contains orthology information about bacteria, archaea and unicellular eukaryotes. The 2018 MBGD version has more than six thousand genomes. The MBGD SPARQL endpoint is available online at http://mbgd.genome.ad.jp/sparql/.

OMA (Altenhoff *et al.*, 2018) provides orthologous gene inferences covering all three domains of life: Archaea, Bacteria, and Eukarya. Although mainly focusing on orthology information, OMA also provides paralogy information (i.e. genes related by duplication). Other homology information is not explicitly available but might be manually or automatically extracted from HOGs (de Farias *et al.*, 2017). The 2018 OMA version has 2167 species and can be queried through the SPARQL endpoint at https://sparql.omabrowser.org/lode/sparql/.

EBI is one of the largest bioinformatics resource providers in Europe (Brooksbank *et al.*, 2014). The EBI RDF platform includes pairwise orthologous genes information from Ensembl database (Zerbino *et al.*, 2018). The SPARQL endpoint to access the EBI data is available at https://www.ebi.ac.uk/rdf/services/sparql. For further details, see https://www.ebi.ac.uk/rdf/documentation/ensembl/.

We group the aforementioned databases based on the orthology information type they provide as follows:

I. **Hierarchical Orthologous Groups (HOGs).** The three data sources that provide evolutionary relationship data in RDF to represent HOGs are **OrthoDB**, **MBGD** and **OMA**. Although the RDF data models of MBGD and OMA both rely on the ORTH ontology (Fernández-Breis *et al.*, 2016), they use different ORTH versions. However, SPARQL queries running over either of the two sources can be formulated in a very similar manner. In the case of OrthoDB, data are organised according to their own internal data model, while also providing cross-references to the UniProt RDF store.

II. **Homologous groups** are sets of homologous genes without any hierarchical grouping ("flat"). All members are homologous to all other members, with no distinction of paralogy or orthology. However, each homologous group can still be associated with a taxonomic level, which indicates to which species clade its members belong. Example of orthology databases from which we can extract these homologous groups are **OMA**, **OrthoDB** and **MBGD**.

III. **Pairwise orthology**. Apart from the aforementioned orthologous groups, evolutionary data can also be provided in the form of pairwise orthologous genes. Among the sources that provide this type of information in RDF, we consider in this article **EBI**, **OMA** and **MBGD**.

Data models

In this section we provide a brief introduction to the data models of the orthology databases considered in this article, in order to facilitate the understanding of the SPARQL queries presented in the Protocol Section.

Figure 1 illustrates a few of the members of a HOG, the main data structure in MBGD. In particular, this MBGD cluster has the identifier *28799*. Members of an MBGD orthologous cluster can be either genes, domains or other clusters. These nested orthologous clusters are built at specific taxonomic levels in the hierarchy. For example, the cluster highlighted in blue in Figure 1 was built at taxonomic level 32, *Myxococcus*. The hierarchy needs to be traversed in order to reach genes, such as *mxa:PL1911* that is highlighted in red in Figure 1, or domains (sub-gene level) which belong to an orthologous cluster at a given taxonomic level.

The RDF model is more suitable for representing such hierarchical data structures than the relational data model (Sima *et al.*, 2019) given that RDF is a graph data model. Moreover, querying orthology RDF data can benefit from SPARQL 1.1 *recursive* graph patterns such as property paths[1]. The main construct in SPARQL required to retrieve the orthologous genes of a gene of interest X will then be the following recursive pattern:

?hog_cluster a orth:OrthologsCluster.

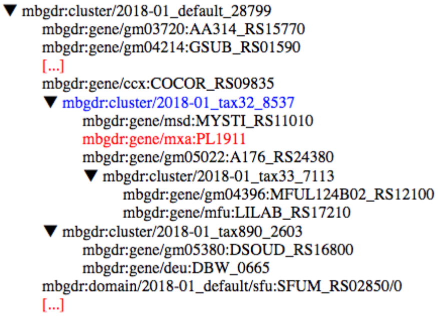
?hog_cluster ***orth:hasHomologous**** *?gene_X.*

?hog_cluster ***orth:hasHomologous**** *?orthologous_gene_Y.*

For example, we can replace *?gene_X* with the URI of the human Hemoglobin Subunit Beta (HBB) gene, namely: <http://mbgd.genome.ad.jp/rdf/resource/gene/hsa:HSA_4504349>, which would enable retrieving all orthologs of human HBB through the *?orthologous_gene_Y* variable. The asterisk (*) following the "*orth:hasHomologous*" property indicates that this property should be matched *recursively*.

A graphical abstraction of the RDF data structure in MBGD is given in Figure 2. SPARQL queries can be formulated by following the direction and labels of arrows in order to formulate triple patterns. For example, to retrieve all genes (i.e. instances of the *orth:Gene* class) of a given HOG, we can follow the graph structure from root to leaf members by performing the query in the following code fragment. In other

[1] https://www.w3.org/TR/sparql11-query/#propertypaths



```
▼ mbgdr:cluster/2018-01_default_28799
    mbgdr:gene/gm03720:AA314_RS15770
    mbgdr:gene/gm04214:GSUB_RS01590
    [...]
    mbgdr:gene/ccx:COCOR_RS09835
  ▼ mbgdr:cluster/2018-01_tax32_8537
        mbgdr:gene/msd:MYSTI_RS11010
        mbgdr:gene/mxa:PL1911
        mbgdr:gene/gm05022:A176_RS24380
      ▼ mbgdr:cluster/2018-01_tax33_7113
            mbgdr:gene/gm04396:MFUL124B02_RS12100
            mbgdr:gene/mfu:LILAB_RS17210
  ▼ mbgdr:cluster/2018-01_tax890_2603
        mbgdr:gene/gm05380:DSOUD_RS16800
        mbgdr:gene/deu:DBW_0665
    mbgdr:domain/2018-01_default/sfu:SFUM_RS02850/0
    [...]
```

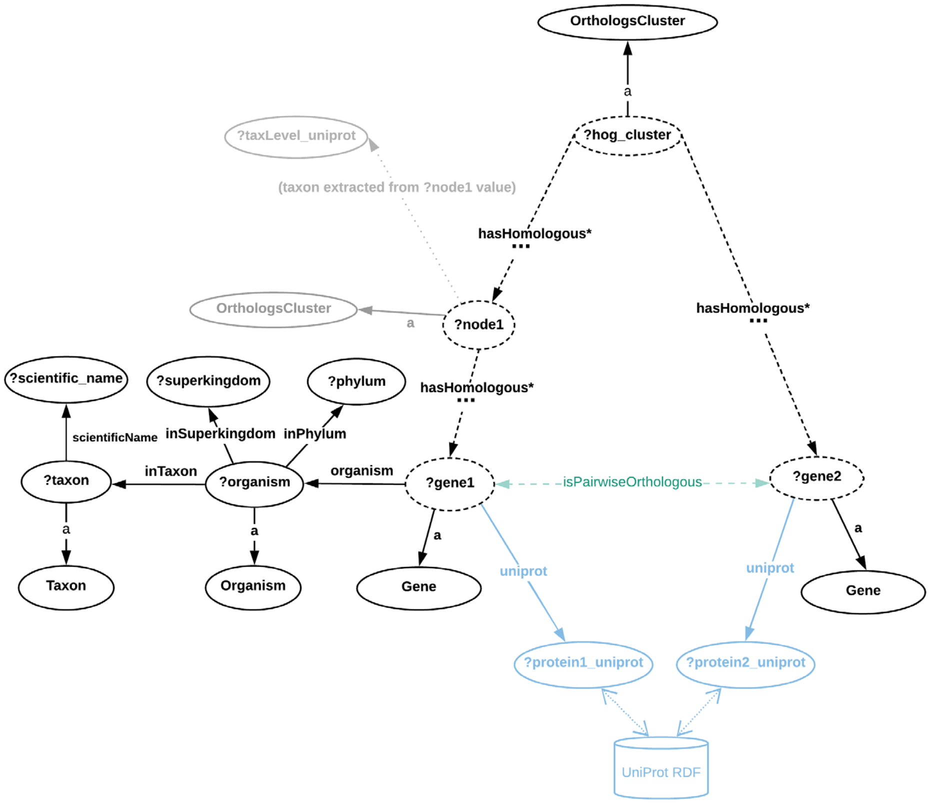
Figure 1. A fragment of the hierarchical orthologous cluster no. 28799 in MBGD. A cluster can consist of genes, domains (sub-genes) or further nested orthologous clusters. Multiple levels of the hierarchy may need to be traversed recursively in order to reach a given orthologous gene. For example, the gene mxa: PL1911 (highlighted in red) can be reached through the member orthologous cluster 2018-01_tax32_8537 (shown in blue). This can be achieved in SPARQL through a recursive graph pattern, using the hasHomologous* property path - a graphical abstraction of the RDF representation is provided in Figure 2.

words, the *?gene1* variable values illustrated as the left-side member in the cluster *?hog_cluster*.

```
PREFIX orth: <http://purl.org/net/orth#>
PREFIX cluster-id:<http://mbgd.genome.ad.jp/rdf/resource/cluster/>


SELECT ?hog_cluster ?gene1 WHERE {
  VALUES ?hog_cluster {cluster-id:2018-01_default_28799}
  ?hog_cluster a orth:OrthologsCluster.
  ?hog_cluster orth:hasHomologous* ?gene1.
  ?gene1 a orth:Gene. }
```

This SPARQL query will retrieve all the genes in the MBGD Hierarchical Orthologous Group represented with the identifier 28799.



e.g.: ?protein1_uniprot = <http://purl.uniprot.org/uniprot/P68871>

Figure 2. Directed-graph abstraction of a portion of the MBGD RDF graph related to hierarchical orthologous groups. Edges are RDF properties and nodes are either classes or variables. The terms preceded by a question mark (e.g. ?gene1) represent variables assigned with either zero or more literals or URIs. Dashed edges illustrate the orth:hasHomologous property that can be stated zero or more times, recursively. URI prefixes were omitted. MBGD is gene-centric and taxonomic ranges where HOGs are built are not directly available in RDF - in some cases these can be extracted from the cluster URI (e.g. http://mbgd.genome.ad.jp/rdf/resource/cluster/2018-01_tax32_8537 corresponds to taxonomic identifier 32, Myxococcus). By contrast, the taxonomic information per gene entry is richer in MBGD than in OMA, including explicit Superkingdom and Phylum information. Example SPARQL queries based on this graph abstraction are provided in the Protocols, as well as in the accompanying Jupyter Notebook. The pairwise orthology information is not directly available (e.g. through an RDF property), but can be extracted from the Orthologs Cluster (to highlight this, the "isPairwiseOrthologous" is shown in green with a dotted arrow).

Similarly, the HOG structure in OMA is abstracted in Figure 3. Both figures can be used as a guide in formulating SPARQL queries, by following the directions of the arrows in order to formulate triple patterns. Since both the MBGD and the OMA models rely on the ORTH ontology (Fernández-Breis *et al.*, 2016), the two graph structures are very similar and therefore SPARQL queries can be formulated with only minor differences for both data sources.

Figure 4 illustrates the data model of the portion of the EBI RDF graph describing orthology information. In contrast to OMA and MBGD, EBI only provides pairwise orthologous genes.

Figure 5 illustrates the structure of Orthologous Groups in the OrthoDB RDF. Here, genes are direct members of OrthoGroups built at a given taxonomic level (Clade), e.g. *Cyanobacteria*. We mention that OrthoDB provides richer information in RDF, including sequence length, number of exons for gene members, as well as evolutionary rates, functional category and others for orthologous groups (for more details see *Extended data*).

Protocols

In this section, we provide four protocols to (i) retrieve pairwise orthologs through SPARQL queries from EBI, OMA, MBGD, as well as (ii) homologous groups from OMA, MBGD and OrthoDB (iii) restrict the search to a given taxonomic level (iv) perform meta-analyses across multiple data sources providing orthology information, as well as aggregations using the entire data available in a given source. All protocols presented below are included in the accompanying Jupyter notebook.

For the sake of simplicity, genes are identified with either their Ensembl identifiers or their cross-reference to the

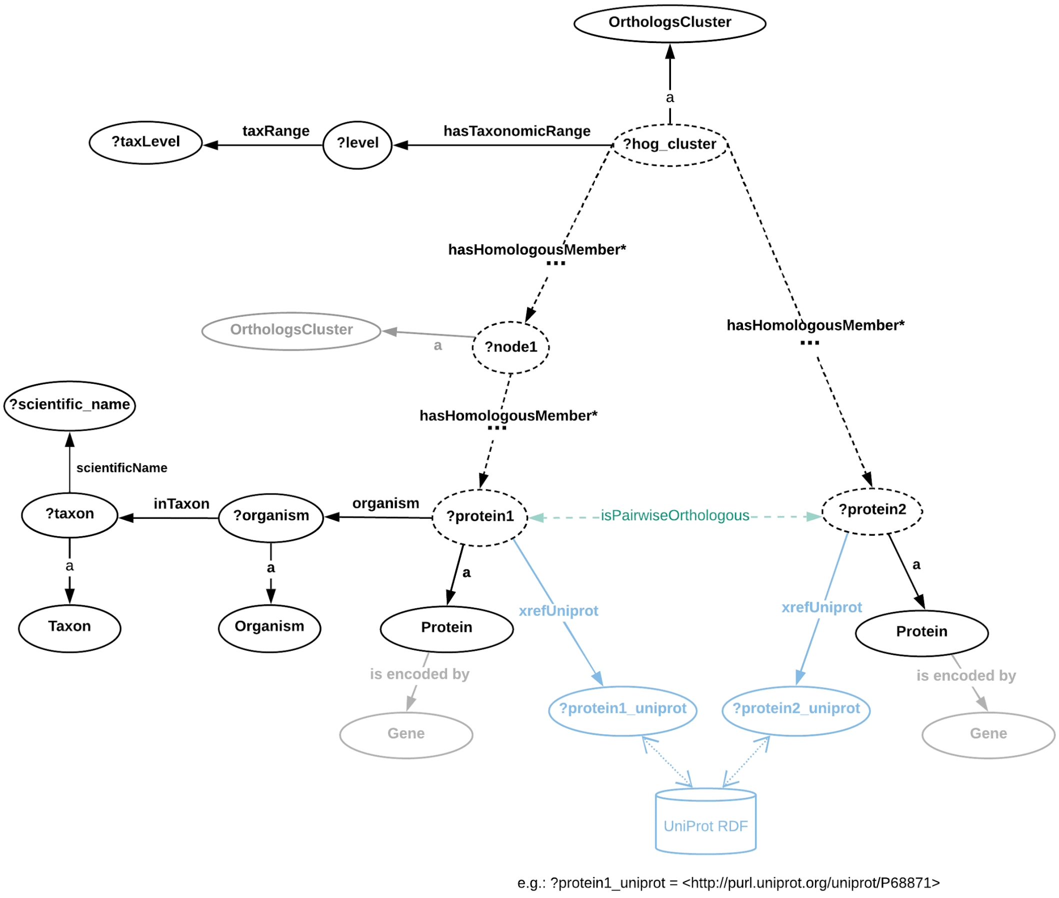
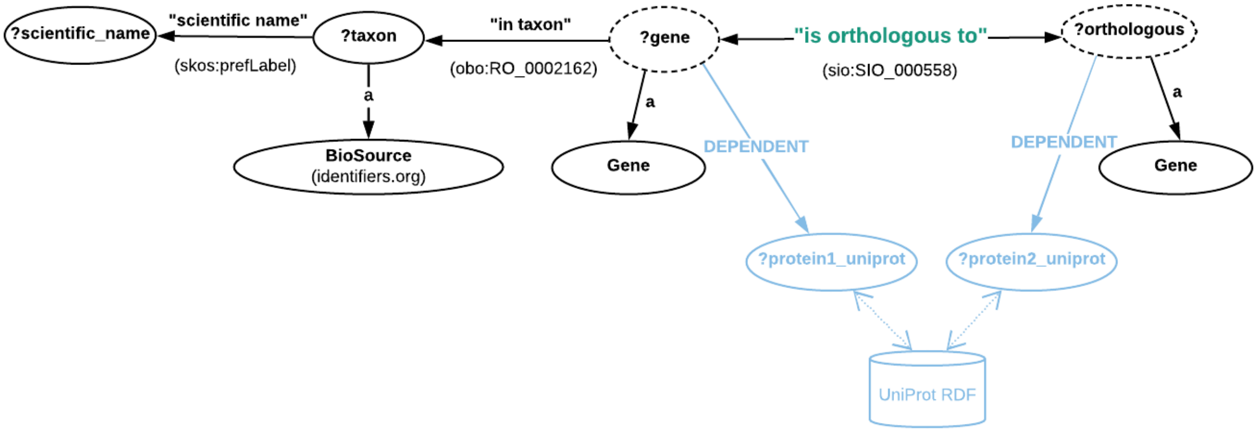
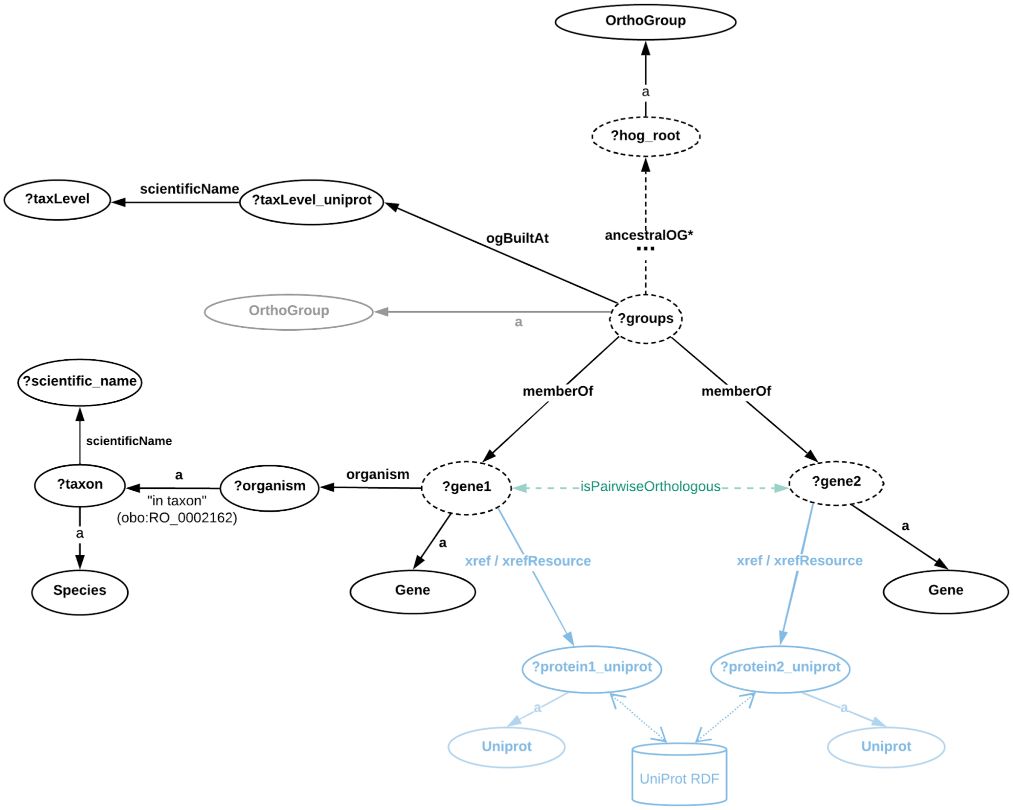


Figure 3. Directed-graph abstraction of a portion of the OMA RDF graph related to hierarchical orthologous groups. Edges are RDF properties and nodes are either classes or variables. The terms preceded by a question mark (e.g. ?protein1) represent variables. Dashed edges illustrate the orth:hasHomologousMember property that can be stated zero or more times, recursively. OMA is protein-centric, however the corresponding genes that encode the proteins are also available in RDF through the “is encoded by” property (a cross-reference to Ensembl identifiers is also provided). Furthermore, the taxonomic ranges where HOGs were built are asserted through the “hasTaxonomicRange” property. The pairwise orthology information is not directly available (e.g. through an RDF property), but can be extracted from the Orthologs Cluster (to highlight this, the “isPairwiseOrthologous” is shown in green with a dotted arrow). Note: URI prefixes were omitted.



e.g.: ?protein1_uniprot = <http://purl.uniprot.org/uniprot/P68871>

Figure 4. Directed-graph abstraction of a portion of the EBI RDF graph related to pairwise orthologous genes. Edges are RDF properties and nodes are either classes or variables. The terms preceded by a question mark (i.e. ?) represent variables assigned with either zero or more literals or URIs. As opposed to the RDF representations in OMA and MBGD, here the pairwise orthology is explicitly asserted through the "is orthologous to" property (more precisely, http://semanticscience.org/resource/SIO_000558). However, there is no information available regarding orthologous clusters. Moreover, the Gene class here is in fact the OBO (not ORTH) class, i.e. http://purl.obolibrary.org/obo/SO_0000704. Instances of these genes can be specified either through their cross-reference to UniProt (the http://rdf.ebi.ac.uk/terms/ensembl/DEPENDENT property) or directly through their ENSEMBL identifier, by fixing the value of ?gene to the concatenation of http://rdf.ebi.ac.uk/resource/ensembl/ and the corresponding Ensembl identifier. Finally, the taxonomic identifiers are provided via instances of the BioSource class, http://www.biopax.org/release/biopax-level3.owl#BioSource.



e.g.: ?protein1_uniprot = <http://purl.uniprot.org/uniprot/P68871>

Figure 5. Directed-graph abstraction of a portion of the OrthoDB RDF graph related to orthologous groups. Edges are RDF properties and nodes are either classes or variables. The terms preceded by a question mark (i.e. ?) represent variables assigned with either zero or more literals or URIs. Genes are direct members of OrthoGroups built at a given taxonomic level (Clade), e.g. Cyanobacteria, available through the ogBuiltAt property. The cross-references to UniProt (as well as Ensembl and Entrez) are available through a 2-triple pattern (for examples see Protocols section).

UniProt accession number. In this article, we assume the reader already knows the UniProt primary accession number of the searched gene. In general, this number can be found by searching for the corresponding gene name in the UniProt webpage, for example, "HBB" (i.e. "hemoglobin subunit beta"). The UniProt protein identifier in RDF is a Uniform Resource Identifier (URI) composed of the UniProt accession number (e.g. P68871) appended to the UniProt namespace prefix: http://purl.uniprot.org/uniprot/. For instance, in the case of the human HBB gene, the URI identifier is http://purl.uniprot.org/uniprot/P68871.

Protocol 1: Retrieve pairwise orthologs (EBI, OMA, MBGD)

In this protocol we illustrate the basic task of retrieving the pairwise orthologs of a given gene, for example the HBB (Hemoglobin subunit beta) human gene. This is illustrated on the three orthology databases that provide pairwise orthology information in RDF: EBI, OMA and MBGD. The corresponding SPARQL queries to retrieve the pairwise orthologs can be formulated as shown below. We note that the resulting orthologs are also provided using their "clickable" cross-reference link to UniProt. This can directly be used to find out more information about the resulting genes (e.g. name, location, expression) and has the added advantage that results originating from different orthology databases can then be compared against each other.

a) Retrieving EBI pairwise orthologs

The following code fragment depicts a SPARQL query to retrieve pairwise orthologs of the human HBB gene from Ensembl dataset at the EBI RDF platform. To execute this query, copy and paste the it into the Web interface of the EBI SPARQL endpoint at https://www.ebi.ac.uk/rdf/services/sparql.

```
PREFIX obo: <http://purl.obolibrary.org/obo/>
PREFIX sio: <http://semanticscience.org/resource/>
PREFIX ensembl: <http://rdf.ebi.ac.uk/resource/ensembl/>
PREFIX ensemblterms: <http://rdf.ebi.ac.uk/terms/ensembl/>

SELECT DISTINCT ?gene_uniprot_uri ?ortholog_uniprot_uri {
  VALUES(?gene_uniprot_uri){(<http://purl.uniprot.org/uniprot/P68871>)}
  ?gene sio:SIO_000558 ?ortholog . # « is orthologous to »
  ?gene obo:RO_0002162 ?taxon . # « in taxon »
  ?ortholog obo:RO_0002162 ?ortholog_taxon.
  ?ortholog ensemblterms:DEPENDENT ?ortholog_uniprot_uri.
  ?gene ensemblterms:DEPENDENT ?gene_uniprot_uri.
  FILTER(?taxon != ?ortholog_taxon
          &&
STRSTARTS(STR(?ortholog_uniprot_uri),"http://purl.uniprot.org/uniprot/”) )}
```

The HBB gene is represented with the UniProt URI http://purl.uniprot.org/uniprot/P68871. To retrieve the orthologs of other genes, we can replace this URI with one that corresponds to another gene such as human INS (i.e. http://purl.uniprot.org/uniprot/P01308). We can also provide a set of URIs enclosed with parentheses such as follows: VALUES(?gene_uniprot_uri) {(<http://purl.uniprot.org/uniprot/P68871>)(<http://purl.uniprot.org/uniprot/P01308>) }. The sio:SIO_000558 is the « is orthologous to » property, while the obo:RO_0002162 represents the « in taxon » property (for a graphical abstraction, see Figure 4).

We note here that not all EBI gene entries have an assigned cross-reference to UniProt. For example, "*ENSG00000139618*" identifies an Ensembl gene for which the UniProt cross-reference is missing from the EBI RDF platform. In this case, the previous SPARQL query can be adapted, by assigning in the VALUES statement of the query, the ?gene variable to the corresponding Ensembl identifier, as depicted in the following code fragment:

```
PREFIX obo: <http://purl.obolibrary.org/obo/>
PREFIX sio: <http://semanticscience.org/resource/>
PREFIX ensembl: <http://rdf.ebi.ac.uk/resource/ensembl/>
PREFIX ensemblterms: <http://rdf.ebi.ac.uk/terms/ensembl/>

SELECT DISTINCT ?gene ?ortholog_uniprot_uri {
  VALUES(?gene){(ensembl:ENSG00000139618)}
  ?gene sio:SIO_000558 ?ortholog.
  ?gene obo:RO_0002162 ?taxon.
  ?ortholog obo:RO_0002162 ?ortholog_taxon.
  ?ortholog ensemblterms:DEPENDENT ?ortholog_uniprot_uri.
  ?gene ensemblterms:DEPENDENT ?gene_uniprot_uri.
FILTER(?taxon != ?ortholog_taxon &&
STRSTARTS(STR(?ortholog_uniprot_uri),"http://purl.uniprot.org/uniprot/”) )}
```

This code fragment illustrates the SPARQL query to retrieve orthologs for the human BRCA2 gene from the Ensembl dataset. The BRCA2 gene is represented with the UniProt URI ensembl:ENSG00000139618 where ensembl is a prefix that replaces http://rdf.ebi.ac.uk/resource/ensembl/. To retrieve the orthologs of other genes, we can replace ensembl:ENSG00000139618 with a URI that corresponds to another gene such as human INS (i.e. ensembl:ENSG00000254647). We can also provide a set of URIs enclosed with parentheses such as follows: VALUES(?gene){(ensembl:ENSG00000139618)(ensembl:ENSG00000254647)}.

b) Retrieving OMA pairwise orthologs

The following code fragment shows a SPARQL query to retrieve pairwise orthologs of the human HBB gene which are derived from the HOGs in the OMA database. To execute the query, copy and paste it in the OMA SPARQL endpoint webpage: https://sparql.omabrowser.org/lode/sparql.

```
PREFIX oma: <http://omabrowser.org/ontology/oma#>
PREFIX orth: <http://purl.org/net/orth#>
PREFIX sio: <http://semanticscience.org/resource/>
PREFIX lscr: <http://purl.org/lscr#>

SELECT DISTINCT ?protein1 ?protein2 {
VALUES(?protein1){(<http://purl.uniprot.org/uniprot/P68871>)}
        ?cluster a orth:OrthologsCluster.
        ?cluster orth:hasHomologousMember ?node1.
        ?cluster orth:hasHomologousMember ?node2.
        ?node1 orth:hasHomologousMember* ?protein_OMA_1.
        ?node2 orth:hasHomologousMember* ?protein_OMA_2.
        ?protein_OMA_1 lscr:xrefUniprot ?protein1.
        ?protein_OMA_2 lscr:xrefUniprot ?protein2.
FILTER(?node1 != ?node2)}
```

The HBB gene is represented in this code fragment with the UniProt URI http://purl.uniprot.org/uniprot/P68871. More precisely, in OMA the lscr:xrefUniprot represents the cross reference to UniProt.

c) Retrieving MBGD pairwise orthologs
In a similar manner to the previous code fragment, the following depicts a SPARQL query to retrieve pairwise orthologs of the human HBB gene which are derived from the HOGs in the MBGD database. To execute the query, copy and paste it in the MBGD SPARQL endpoint webpage: http://mbgd.genome.ad.jp/sparql/.

```
PREFIX mbgdr: <http://mbgd.genome.ad.jp/rdf/resource/>
PREFIX mbgd: <http://purl.jp/bio/11/mbgd#>
PREFIX orth: <http://purl.org/net/orth#>

SELECT ?protein1 ?protein2
WHERE {
VALUES(?protein1){ (<http://purl.uniprot.org/uniprot/P68871>)}
        ?cluster a orth:OrthologsCluster.
        ?cluster orth:hasHomologous ?node1.
        ?cluster orth:hasHomologous ?node2.
        ?node1 orth:hasHomologous* ?gene1.
        ?node2 orth:hasHomologous* ?gene2.
        ?gene1 mbgd:uniprot ?protein1.
        ?gene2 mbgd:uniprot ?protein2.
FILTER(?node1 != ?node2)}
```

The HBB gene is represented again with the UniProt URI http://purl.uniprot.org/uniprot/P68871. In the case of MBGD, the mbgd:uniprot represents the cross-reference to UniProt.

Protocol 2: Retrieve homologous groups

In this protocol we illustrate the task of retrieving the non-hierarchical homologous groups of a target gene, such as the human HBB gene. In addition, we restrict the search to a specific taxonomic level, for example, "only at the primates level". In other words, we depict how to retrieve the homologous groups at a given taxonomic level and including a given gene represented as a UniProt URI. Note that the same query can be executed only providing one of the inputs (i.e. either the taxonomic level or gene). However, it can take longer to return all results or may not even be executed due to runtime constraints at the original databases. The members of a homologous group can be either paralogous or orthologous to one another.

a) Retrieving OMA Homologous Groups derived from the HOGs
The following code fragment can be executed at the OMA SPARQL endpoint webpage at https://sparql.omabrowser.org/lode/sparql.

```
PREFIX lscr: <http://purl.org/lscr#>
PREFIX orth: <http://purl.org/net/orth#>

SELECT DISTINCT ?cluster ?protein2_OMA_URI ?protein2_uniprot_URI
?tax_name {
 VALUES(?protein1_uniprot_URI){(<http://purl.uniprot.org/uniprot/P68871>)}
 VALUES(?tax_name){("Primates")}
 ?cluster a orth:OrthologsCluster.
 ?cluster orth:hasHomologousMember* ?protein_OMA_1.
 ?cluster orth:hasHomologousMember* ?protein2_OMA_URI.
 ?protein_OMA_1 a orth:Protein.
 ?protein2_OMA_URI a orth:Protein.
 ?protein_OMA_1 lscr:xrefUniprot ?protein1_uniprot_URI.
 OPTIONAL{?protein2_OMA_URI lscr:xrefUniprot ?protein2_uniprot_URI.}
 ?cluster orth:hasTaxonomicRange ?tax.
 ?tax orth:taxRange ?tax_name. }
```

This code fragment illustrates the SPARQL query to retrieve homologous groups (i.e. clusters) that contains the human HBB gene in the OMA database. The HBB gene is represented with its related UniProt entry (i.e. the UniProt URI http://purl.uniprot.org/uniprot/P68871). To retrieve the clusters that have other genes, we can replace this URI with one that corresponds to another gene such as human INS (i.e. http://purl.uniprot.org/uniprot/P01308). We can also provide a set of URIs enclosed with parentheses such as follows:

VALUES(?protein1_uniprot_URI) {(<http://purl.uniprot.org/uniprot/P68871>) (<http://purl.uniprot.org/uniprot/P01308>)}. Similarly, we can change the taxonomic level of reference as follows: VALUES(?tax_name) {("Hominoidea")}.

b) MBGD Homologous Groups derived from the HOGs
The HOGs in MBGD do not provide explicit taxonomic levels at the root level of a HOG. However, the taxon NCBI identifiers of subHOGs (i.e. sublevels) can be extracted in some cases from the cluster URI. Since this requires more advanced knowledge of SPARQL (in particular, for parsing the cluster URIs), we only make it available as part of the *Extended data*.

c) OrthoDB Homologous Groups

The following code can be executed at the OrthoDB SPARQL endpoint webpage at https://sparql.orthodb.org.

```
PREFIX orthodb: <http://purl.orthodb.org/>
PREFIX up: <http://purl.uniprot.org/core/>

SELECT DISTINCT ?groups ?species_name ?protein1_uniprot ?gene1
?taxLevel_uniprot ?taxLevel WHERE {
        VALUES ?protein2_uniprot {<http://purl.uniprot.org/uniprot/P68871>}
        VALUES ?taxLevel {"Primates"}

        ?gene2 a orthodb:Gene.
        ?gene2 orthodb:memberOf ?groups.
        ?gene1 a orthodb:Gene.
        ?gene1 orthodb:memberOf ?groups.
        ?gene1 up:organism ?organism.
        ?organism a ?taxon.
        ?taxon up:scientificName ?species_name.
        ?groups orthodb:ogBuiltAt ?taxLevel_uniprot.
        ?taxLevel_uniprot up:scientificName ?taxLevel.
        ?gene2 orthodb:xref ?xref2.
        ?xref2 orthodb:xrefResource ?protein2_uniprot.
        ?protein2_uniprot a orthodb:Uniprot.
        ?gene1 orthodb:xref ?xref.
        ?xref a orthodb:Xref.
        OPTIONAL{
        ?xref orthodb:xrefResource ?protein1_uniprot.
        ?protein1_uniprot a orthodb:Uniprot.}
} ORDER BY ?groups, ?taxLevel
```

This SPARQL query will retrieve flat homologous groups (i.e. clusters) that contains the human HBB gene in OrthoDB. The HBB gene is represented with its related UniProt entry (i.e. the UniProt URI http://purl.uniprot.org/uniprot/P68871).

Protocol 3: Retrieve Hierarchical Orthologous Groups (HOG)

In this protocol we show how to retrieve the HOGs containing a target gene, such as the human HBB gene, in the three orthology databases OMA, MBGD and OrthoDB. The Ensembl dataset in the EBI RDF platform is not considered because it does not provide HOG information.

a) Retrieving HOGs from OMA

The following code fragment can be executed at the OMA SPARQL endpoint webpage at https://sparql.omabrowser.org/lode/sparql.

```
PREFIX obo: <http://purl.obolibrary.org/obo/>
PREFIX orth: <http://purl.org/net/orth#>
PREFIX taxon: <http://purl.uniprot.org/taxonomy/>
PREFIX up: <http://purl.uniprot.org/core/>
PREFIX lscr: <http://purl.org/lscr#>

SELECT DISTINCT ?root_hog ?species_name ?protein1_uniprot
(?protein1 as
                        ?protein1_OMA) ?taxLevel {
        VALUES ?protein2_uniprot {<http://purl.uniprot.org/uniprot/P68871>}
        ?root_hog obo:CDAO_0000148 ?hog_cluster. #has_Root
        ?hog_cluster orth:hasHomologousMember* ?node1.
        ?node1 a orth:OrthologsCluster.
        ?node1 orth:hasTaxonomicRange ?level.
        ?level orth:taxRange ?taxLevel.
        ?node1 orth:hasHomologousMember* ?protein1.
        ?hog_cluster orth:hasHomologousMember* ?protein2.
        ?protein1 a orth:Protein.
        ?protein1 orth:organism ?organism.
        ?organism obo:RO_0002162 ?taxon.
        ?taxon up:scientificName ?species_name.
        OPTIONAL{?protein1 lscr:xrefUniprot ?protein1_uniprot}.
        ?protein2 a orth:Protein.
        ?protein2 lscr:xrefUniprot ?protein2_uniprot.
} ORDER BY ?taxLevel
```

This SPARQL query will retrieve hierarchical orthologous groups that contain the human HBB gene in the OMA database. The HBB gene is represented with its related UniProt entry (i.e. the UniProt URI http://purl.uniprot.org/uniprot/P68871).

b) Retrieving HOGs from MBGD

The SPARQL query to retrieve HOGs from MBGD is similar to the previous query over OMA and therefore we make it available as *Extended data*. As a reminder, although both the OMA and MBGD databases rely on different versions of the ORTH ontology, they structure their HOG data similarly.

c) Retrieving HOGs from OrthoDB

The following code fragment can be executed at the OrthoDB SPARQL endpoint webpage at https://sparql.orthodb.org. Note that the OrthoDB HOGs often do not contain all taxonomic levels between the orthologous cluster at the highest taxonomic level (i.e. the root) and genes (i.e. leaves).

```
PREFIX orthodb: <http://purl.orthodb.org/>
PREFIX up: <http://purl.uniprot.org/core/>

SELECT DISTINCT ?hog_root ?species_name ?protein1_uniprot ?gene1
?taxLevel_uniprot ?taxLevel
WHERE {
VALUES ?protein2_uniprot {<http://purl.uniprot.org/uniprot/P68871>}

?gene2 a orthodb:Gene.
?gene2 orthodb:memberOf ?groups.
?gene2 orthodb:memberOf ?hog_root.
FILTER NOT EXISTS {?hog_root orthodb:ancestralOG ?ancestor.}
?groups orthodb:ancestralOG* ?hog_root.
?gene1 a orthodb:Gene.
?gene1 orthodb:memberOf ?groups.
?gene1 up:organism ?organism.
?organism a ?taxon.
?taxon up:scientificName ?species_name.
?groups orthodb:ogBuiltAt ?taxLevel_uniprot.
?taxLevel_uniprot up:scientificName ?taxLevel.
?gene2 orthodb:xref ?xref2.
?xref2 orthodb:xrefResource ?protein2_uniprot.
?protein2_uniprot a orthodb:Uniprot.
?gene1 orthodb:xref ?xref.
?xref orthodb:xrefResource ?protein1_uniprot.
?protein1_uniprot a orthodb:Uniprot.
} ORDER BY ?hog_root, ?taxLevel
```

This SPARQL query will retrieve hierarchical orthologous groups that contains the human HBB gene in the OrthoDB database. The HBB gene is represented with its related UniProt entry (i.e. the UniProt URI http://purl.uniprot.org/uniprot/P68871).

Protocol 4: Meta-analysis - comparing data across OMA and MBGD orthology

In this protocol, we show how to compare orthology information across multiple databases with SPARQL 1.1. Although the example in the following code fragment is restricted to OMA and MBGD, similar queries over different combinations of the orthology databases mentioned in this article can be derived based on the Code Fragments in Protocols 1, 2 and 3.

For a given UniProt entry such as the accession number K9Z723, retrieve orthologs that are **only** in MBGD, but not in OMA. Alternatively, to retrieve only those that appear in **both** sources, simply remove the "NOT" keyword in the FILTER clause below. To execute the query, copy and paste it in the SPARQL endpoint page of OMA: https://sparql.omabrowser.org/lode/sparql.

```
PREFIX oma: <http://omabrowser.org/ontology/oma#>
PREFIX orth: <http://purl.org/net/orth#>
PREFIX sio: <http://semanticscience.org/resource/>
PREFIX lscr: <http://purl.org/lscr#>
PREFIX mbgd: <http://purl.jp/bio/11/mbgd#>

SELECT ?protein2 ?species WHERE {
 SERVICE<http://sparql.nibb.ac.jp/sparql> {
  SELECT ?protein2 ?species where {
          ?cluster_mbgd a orth:OrthologsCluster.
          ?cluster_mbgd orth:hasHomologous ?node1_mbgd.
          ?cluster_mbgd orth:hasHomologous ?node2_mbgd.
          ?node1_mbgd orth:hasHomologous* ?gene1.
          ?node2_mbgd orth:hasHomologous* ?gene2.
          ?gene1 mbgd:uniprot <http://purl.uniprot.org/uniprot/K9Z723>.
          ?gene2 mbgd:uniprot ?protein2.
          ?gene2 mbgd:organism ?taxon.
          OPTIONAL{?taxon mbgd:species ?species.}
          FILTER(?node1_mbgd != ?node2_mbgd) } }
 FILTER NOT EXISTS { # keep only those that do not exist in OMA
     ?cluster a orth:OrthologsCluster.
     ?cluster orth:hasHomologousMember ?node1.
     ?cluster orth:hasHomologousMember ?node2.
     ?node1 orth:hasHomologousMember* ?protein_OMA_1.
     ?node2 orth:hasHomologousMember* ?protein_OMA_2.
     ?protein_OMA_1 lscr:xrefUniprot <http://purl.uniprot.org/uniprot/K9Z723>.
     ?protein_OMA_2 lscr:xrefUniprot ?protein2.
     FILTER(?node1 != ?node2) }}
```

This federated SPARQL query will retrieve pairwise orthologous genes of the Cyanobacterium-aponinum psb27- gene that are found in the MBGD database but are not present in OMA. The psb27 gene is represented with its related UniProt entry, thus the UniProt URI http://purl.uniprot.org/uniprot/K9Z723.

Aggregations in SPARQL: Combining data from multiple resources

In the *Extended data*, we provide additional examples showing how to retrieve the top 10 entries with most orthologs in OMA and MBGD for a given species, e.g. *'Drosophila melanogaster'*. These examples illustrate a few more advanced SPARQL features, such as aggregation and ordering by a criterion in order to select the top N results.

Conclusion

We provide four protocols that show how to query evolutionary relationships (pairwise orthologs, as well as HOGs) across four major databases available through SPARQL 1.1 endpoints: EBI, OMA, MBGD and OrthoDB. The protocols presented can serve as a useful starting point for readers interested in an introduction to the RDF data models of these sources, as well as the basics of retrieving orthology information through SPARQL queries. Finally, we have shown how aggregations in SPARQL can be used to quickly generate an overview of the data available in each considered database, and how this data can be compared across the sources.

To sum up, we hope these protocols provide a useful introduction into analysing evolutionary relationships among genes with SPARQL, as well as enriching these analyses by integrating information from external sources, through federated queries. We encourage readers to experiment with the examples presented in this article, which are provided in the accompanying Jupyter notebook, to be directly re-used or integrated into existing research analysis pipelines. As future work, we plan to integrate the queries in this protocol in the BioQuery search interface (Sima *et al.*, 2019) already available for OMA data at http://biosoda.expasy.org/ in order to enable researchers to directly execute or further refine them in a user-friendly environment.

Data availability

Underlying data

Protocols available from: https://github.com/biosoda/tutorial_orthology/blob/master/Orthology_SPARQL_Notebook.ipynb

Archived protocols as at time of publication: http://doi.org/10.5281/zenodo.3499928

License: CC0

Extended data

Zenodo: Protocols to retrieve orthology information with SPARQL, http://doi.org/10.5281/zenodo.3499928 (Sima & Mendes de Farias, 2019).

This project contains the following extended data:

- Table 1. “Cheat sheet” for RDF data available in the four sources considered in this tutorial. (*) GO annotations can be retrieved from the UniProt RDF store through UniProt cross-references.
- Supplementary protocols: Retrieving MBGD Homologous Groups
- Aggregation queries

Data are available under the terms of the Creative Commons Zero “No rights reserved” data waiver (CC0 1.0 Public domain dedication).

Software availability

SPARQL endpoints (all links include further example queries):

- ***OMA*** https://sparql.omabrowser.org/lode/sparql
- ***OrthoDB*** https://sparql.orthodb.org/
- ***EBI*** https://www.ebi.ac.uk/rdf/services/sparql
    - for example orthology queries see “Ensembl” category
- ***MBGD*** http://mbgd.genome.ad.jp/sparql/

References

Altenhoff AM, Gil M, Gonnet GH, *et al.*: **Inferring hierarchical orthologous groups from orthologous gene pairs.** *PLoS One.* 2013; **8**(1): e53786.
PubMed Abstract | Publisher Full Text | Free Full Text

Altenhoff AM, Glover NM, Train CM, *et al.*: **The OMA orthology database in 2018: retrieving evolutionary relationships among all domains of life through richer web and programmatic interfaces.** *Nucleic Acids Res.* 2018; **46**(D1): 477–485.
PubMed Abstract | Free Full Text

Brooksbank C, Bergman MT, Apweiler R, *et al.*: **The European Bioinformatics Institute’s data resources 2014.** *Nucleic Acids Res.* 2014; **42**(Database issue): D18–25.
PubMed Abstract | Publisher Full Text | Free Full Text

Chiba H, Nishide H, Uchiyama I: **Construction of an ortholog database using the semantic web technology for integrative analysis of genomic data.** *PLoS One.* 2015; **10**(4): e0122802.
PubMed Abstract | Publisher Full Text | Free Full Text

de Farias TM, Chiba H, Fernández-Breis JT: **Leveraging logical rules for efficacious representation of large orthology datasets**. s.l., *Proceedings of the 10th International Semantic Web Applications and Tools for Healthcare and Life Sciences (SWAT4HCLS) Conference.* 2017.
Reference Source

Duek P, Gateau A, Bairoch A, *et al.*: **Exploring the Uncharacterized Human Proteome Using neXtProt.** *J Proteome Res.* 2018; **17**(12): 4211–4226.
PubMed Abstract | Publisher Full Text

Fernández-Breis JT, Chiba H, Legaz-García Mdel C, *et al.*: **The Orthology Ontology: development and applications.** *J Biomed Semantics.* 2016; **7**(1): 34.
PubMed Abstract | Publisher Full Text | Free Full Text

Fitch WM: **Distinguishing homologous from analogous proteins.** *Syst Zool.* 1970; **19**(2): 99–113.
PubMed Abstract | Publisher Full Text

Gabaldón T, Koonin EV: **Functional and evolutionary implications of gene orthology.** *Nat Rev Genet.* 2013; **14**(5): 360–6.
PubMed Abstract | Publisher Full Text | Free Full Text

Glover N, Dessimoz C, Ebersberger I, *et al.*: **Advances and Applications in the Quest for Orthologs.** *Mol Biol Evol.* 2019; **36**(10): 2157–2164.
PubMed Abstract | Publisher Full Text

Iyappan A, Kawalia SB, Raschka T, *et al.*: **NeuroRDF: semantic integration of highly curated data to prioritize biomarker candidates in Alzheimer’s disease.** *J Biomed Semantics.* 2016; **7**: 45.
PubMed Abstract | Publisher Full Text | Free Full Text

Jupp S, Malone J, Bolleman J, *et al.*: **The EBI RDF platform: linked open data for the life sciences.** *Bioinformatics.* 2014; **30**(9): 1338–9.
PubMed Abstract | Publisher Full Text | Free Full Text

Sima AC, De Farias TM, Zbinden E, *et al.*: **Enabling Semantic Queries Across Federated Bioinformatics Databases.** *Database (to appear).* 2019.
Publisher Full Text

Sima AC, Mendes de Farias T: **Protocols to retrieve orthology information with SPARQL (Version v1.0.0-beta).** *Zenodo.* 2019.
http://www.doi.org/10.5281/zenodo.3499928

Sima AC, Stockinger K, de Farias TM, *et al.*: **Semantic integration and enrichment of heterogeneous biological databases**. *Methods Mol Biol.* In: *Evolutionary Genomics.* s.l.: Springer, 2019; **1910**: 655–690.
PubMed Abstract | Publisher Full Text

W3C SPARQL WORKING GROUP, *et al.*: **SPARQL 1.1 overview.** W3C

recommendation. World Wide Web Consortium, Cambridge, MA, USA. (accessed 24/10/2019), 2013.
Reference Source

Uchiyama I, Mihara M, Nishide H, *et al.*: **MBGD update 2018: microbial genome database based on hierarchical orthology relations covering closely related and distantly related comparisons.** *Nucleic Acids Res.* 2018; **47**(D1): 382–389.
Publisher Full Text

Williams AJ, Harland L, Groth P, *et al.*: **Open PHACTS: semantic interoperability for drug discovery.** *Drug Discov Today.* 2012; **17**(21–22): 1188–1198.
PubMed Abstract | Publisher Full Text

Zdobnov EM, Tegenfeldt F, Kuznetsov D, *et al.*: **OrthoDB v9.1: cataloging evolutionary and functional annotations for animal, fungal, plant, archaeal, bacterial and viral orthologs.** *Nucleic Acids Res.* 2017; **45**(D1): 744–649.
PubMed Abstract | Publisher Full Text | Free Full Text

Zerbino DR, Achuthan P, Akanni W, *et al.*: **Ensembl 2018.** *Nucleic Acids Res.* 2018; **46**(D1): 754–761.
PubMed Abstract | Publisher Full Text | Free Full Text

Open Peer Review

Current Peer Review Status: ? ? ✓

Version 1

Reviewer Report 29 June 2020

https://doi.org/10.5256/f1000research.23139.r64153

✓

Tatsuya Kushida
[1] National Bioscience Database Center, Tokyo, Japan
[2] RIKEN BioResource Research Center, Tsukuba, Japan

It can be useful to leverage orthologous gene knowledge in genome data analysis like gene function annotations, and the inference. The authors attempted to introduce how to retrieve the orthologous genes for several RDF databases such as OMA, MBGD, OrthoDB, and EnsemblRDF, using concrete SPARQL query examples. In particular, I highly evaluate that the authors explain through appropriate examples of the methods to retrieve the information of the complicated orthologous structures. It is expected to facilitate the understanding of the knowledge, and it would make contributions to increase the usage for genome data analysis. Moreover, the authors provided the Python codes' information to extract orthologous genes from the RDF databases and incorporate them into existing analysis pipelines. That is also worth appreciating. It is essential to further appeal the merits to bioinformaticians and experimental researchers in order to disseminate the orthologous RDF data.

I realized that as stated in the section: Aggregations in SPARQL: Combining data from multiple resources (page 11), I can aggregate the data more efficiently using the SPARQL than traditional procedure (file-based data exchange or full database dumps.) On the other hand, the authors did not sufficiently mention the advantage of the RDF and SPARQL compared with the traditional procedure. I hope that the authors would more strongly emphasize the advantage.

Is the rationale for developing the new method (or application) clearly explained?

Yes

Is the description of the method technically sound?

Partly

Are sufficient details provided to allow replication of the method development and its use by others?

Partly

If any results are presented, are all the source data underlying the results available to ensure full reproducibility?
Yes

Are the conclusions about the method and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Bioinformatics, Database integration, RDF, Ontology

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Author Response 08 Jul 2020
Tarcísio Mendes, University of Lausanne, Lausanne, Switzerland

Dear Tatsuya Kushida,

We would like to thank you for the insightful comments and suggestions for improvements. We have revised the article in order to take all comments into account. We thus hope that this new version takes into account the remarks and answers the questions raised.

In summary, we have made the following changes:

- We have added all the queries presented in the article to the BioQuery3 interface online, (available at biosoda.expasy.org), Section "Information on Homologous Gene queries", making it easier for readers to try them out directly without requiring copy-pasting the SPARQL queries from the text of the article. Furthermore, we have also created clickable links that point users directly to the corresponding query results in the relevant SPARQL endpoint webpage, in the form of purl.org/orthology/q0 to purl.org/orthology/q9, where the ids Q0-Q9 correspond to the queries presented in the “Protocols” section.
- We have added a subsection "Applicable Ontologies", in the "Materials" section, explaining the ontologies relevant to the orthology domain, used by the four data sources described in the article (OMA, OrthoDB, MBGD and EBI).
- We have improved the explanations regarding the choice of relevant IRIs for formulating SPARQL queries targeting the orthology domain, while clarifying the specific challenges of querying orthology data with SPARQL.
- We have improved the OrthoDB data model figure with small corrections.
- We have included a subsection about “retrieving OrthoDB pairwise orthologs”.

The following paragraphs list the issues identified by the reviewers, along with our corresponding answers. Below we provide detailed answers to your comments.

It can be useful to leverage orthologous gene knowledge in genome data analysis like gene function annotations, and the inference. The authors attempted to introduce how to retrieve the orthologous genes for several RDF databases such as OMA, MBGD, OrthoDB, and EnsemblRDF, using concrete SPARQL query examples. In particular, I highly evaluate that the authors explain through appropriate examples of the methods to retrieve the information of the complicated *orthologous structures. It is expected to facilitate the understanding of the knowledge, and it would make contributions to increase the usage for genome data analysis. Moreover, the authors provided the Python codes' information to extract orthologous genes from the RDF databases and incorporate them into existing analysis pipelines. That is also worth appreciating. It is essential to further appeal the merits to bioinformaticians and experimental researchers in order to disseminate the orthologous RDF data.*

- *I realized that as stated in the section: Aggregations in SPARQL: Combining data from multiple resources (page 11), I can aggregate the data more efficiently using the SPARQL than traditional procedure (file-based data exchange or full database dumps.) On the other hand, the authors did not sufficiently mention the advantage of the RDF and SPARQL compared with the traditional procedure. I hope that the authors would more strongly emphasize the advantage.*

Authors: We now included a better explanation to emphasize this advantage in the subsection “Aggregations in SPARQL: Combining data from multiple resources”.

Competing Interests: No competing interests were disclosed.

Reviewer Report 12 June 2020

https://doi.org/10.5256/f1000research.23139.r55913

? **Ikuo Uchiyama**

Laboratory of Genome Informatics, National Institute for Basic Biology, National Institutes of Natural Sciences, Okazaki, Japan

In this manuscript, the authors provide introductory examples showing how to query hierarchical orthology databases in RDF format using the SPARQL query language. Although these semantic web technologies are promising for integrating various biological resources, they still seem to be utilized by only very limited researchers. In this respect, this kind of introductory paper may be helpful for more biologists to utilize these technologies. However, for this purpose, I think that this manuscript has a serious defect that it is not easily understandable unless the readers have sufficient knowledge and skills of BOTH SPARQL and homology/orthology concepts (I do not think that this article contains any specific new ideas for such experts). Therefore, the authors should add some more explanation to improve the understandability for those who are not very familiar with either of these fields in order to achieve the goal of this article.

1. At the first example in "Data models" section (page 4), the authors should add some explanation about how to make a query based on a given RDF graph (with some figure). Such an explanation in addition to the RDF graphs of the actual databases can help understand the subsequent SPARQL codes.

2. Here, protocols are shown for extracting three types of homology/orthology information: pairwise orthologs, homologous groups, and hierarchical orthologous groups (HOGs), but the exact definitions of the extracted information are not clearly given. Moreover, the specifications of the protocols including their purposes and outcomes are not clearly explained. In fact, all of these

protocols retrieve some kinds of pairwise homology/orthology relationships of a given UniProt entry, but the differences of them are not very obvious. Therefore, it is not easy for readers to understand these protocols exactly, and even impossible to check the correctness of the code, without any appropriate explanation.

3. Related but somewhat minor comment: homologous group considered here seems to correspond to orthologous group that contains only orthologs and in-paralogs, while I think homologous group can generally also contain out-paralogs.

4. Most readers are probably not satisfied with just a copy-and-paste of the given examples, but want to modify them for their own purposes, so it is desirable to give some hints on how to make modifications to each query.

5. Some codes containing double quotation marks did not work well when I did copy and paste them from the article web site, probably because these characters were converted into non-ASCII characters.

6. In the last example, the URIs of the same UniProt entry are appeared repeatedly. Can’t you use a common variable to specify the query URI only once?

Is the rationale for developing the new method (or application) clearly explained?
Yes

Is the description of the method technically sound?
Partly

Are sufficient details provided to allow replication of the method development and its use by others?
Partly

If any results are presented, are all the source data underlying the results available to ensure full reproducibility?
Yes

Are the conclusions about the method and its performance adequately supported by the findings presented in the article?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Bioinformatics, specifically on comparative and evolutionary genomics and genome databases.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 08 Jul 2020

Tarcísio Mendes, University of Lausanne, Lausanne, Switzerland

Dear Ikuo Uchiyama,

We would like to thank you for the insightful comments and suggestions for improvements. We have revised the article in order to take all comments into account. We thus hope that this new version takes into account the remarks and answers the questions raised.

In summary, we have made the following changes:

- We have added all the queries presented in the article to the BioQuery3 interface online, (available at biosoda.expasy.org), Section "Information on Homologous Gene queries", making it easier for readers to try them out directly without requiring copy-pasting the SPARQL queries from the text of the article. Furthermore, we have also created clickable links that point users directly to the corresponding query results in the relevant SPARQL endpoint webpage, in the form of purl.org/orthology/q0 to purl.org/orthology/q9, where the ids Q0-Q9 correspond to the queries presented in the “Protocols” section.
- We have added a subsection "Applicable Ontologies", in the "Materials" section, explaining the ontologies relevant to the orthology domain, used by the four data sources described in the article (OMA, OrthoDB, MBGD and EBI).
- We have improved the explanations regarding the choice of relevant IRIs for formulating SPARQL queries targeting the orthology domain, while clarifying the specific challenges of querying orthology data with SPARQL.
- We have improved the OrthoDB data model figure with small corrections.
- We have included a subsection about “retrieving OrthoDB pairwise orthologs”.

The following paragraphs list the issues identified by you, along with our corresponding answers. Below we provide detailed answers to your comments.

In this manuscript, the authors provide introductory examples showing how to query hierarchical orthology databases in RDF format using the SPARQL query language. Although these semantic web technologies are promising for integrating various biological resources, they still seem to be utilized by only very limited researchers. In this respect, this kind of introductory paper may be helpful for more biologists to utilize these technologies.

- *However, for this purpose, I think that this manuscript has a serious defect that it is not easily understandable unless the readers have sufficient knowledge and skills of BOTH SPARQL and homology/orthology concepts (I do not think that this article contains any specific new ideas for such experts). Therefore, the authors should add some more explanation to improve the understandability for those who are not very familiar with either of these fields in order to achieve the goal of this article.*

Authors: To improve understandability we have added better explanations and citations throughout this article (see further details in the answers below).

- *At the first example in "Data models" section (page 4), the authors should add some explanation about how to make a query based on a given RDF graph (with some figure). Such an explanation in addition to the RDF graphs of the actual databases can help understand the subsequent SPARQL codes.*

Authors: We have added a new figure in the beginning of the “Data Models” section as well as an explanation regarding how to write a basic SPARQL query starting from this (simplified) graph model. We now also refer readers to an introductory book chapter we recently published on the topic.

- *Here, protocols are shown for extracting three types of homology/orthology information: pairwise orthologs, homologous groups, and hierarchical orthologous groups (HOGs), but the exact definitions of the extracted information are not clearly given. Moreover, the specifications of the protocols including their purposes and outcomes are not clearly explained. In fact, all of these protocols retrieve some kinds of pairwise homology/orthology relationships of a given UniProt entry, but the differences of them are not very obvious. Therefore, it is not easy for readers to understand these protocols exactly, and even impossible to check the correctness of the code, without any appropriate explanation.*

Authors: We agree that the terminology is confusing. In fact, in the meantime, we have published a separate F1000 paper which introduces the various types of orthologs used in OMA (https://f1000research.com/articles/9-27). We now refer to this, and have also added better explanations of the retrieved results for each query type (e.g. pairwise orthologs, homologous groups, and HOGs). To check the correctness of the retrieved results, the reader has to rely on potentially more time consuming alternative solutions provided by each independent database (e.g. RESTful APIs, Python libraries) or perform interactive searches on the omabrowser.org website.

- *Related but somewhat minor comment: homologous group considered here seems to correspond to orthologous group that contains only orthologs and in-paralogs, while I think homologous group can generally also contain out-paralogs.*

Authors: The reviewer is correct. We have added a sentence of explanation ("However, the kind of homologous groups considered in this tutorial do not contain "out-paralogs"—i.e. paralogs which result from gene duplications which took place prior to the last common ancestor of all species in the databases").

- *Most readers are probably not satisfied with just a copy-and-paste of the given examples, but want to modify them for their own purposes, so it is desirable to give some hints on how to make modifications to each query.*

Authors: We have added an introduction on how to formulate a SPARQL query starting from an example data model graph, also pointing to the Ontology Lookup Service for mapping terms to their RDF identifiers (IRIs). We also indicate to the reader how to formulate SPARQL queries starting from the first data model introduced, MBGD, by following the direction of the arrows in the simplified data model figure. Finally, we have also added a dedicated subsection "Choosing the relevant target gene identifiers in RDF" explaining how to change the examples provided with the corresponding target genes of interest. Changing the queries should be facilitated by the availability of direct clickable links to the queries (see explanation below).

- *Some codes containing double quotation marks did not work well when I did copy and paste them from the article web site, probably because these characters were converted into non-ASCII characters.*

Authors: To solve this problem, we have now created clickable links that point users directly to the corresponding query results in the relevant SPARQL endpoint webpage. This avoids potential errors resulting from copy-pasting the formatted text into the relevant SPARQL endpoint webpage where the query is executed or possibly further modified.
Furthermore, we have added all the queries presented in this article also to the BioQuery3 interface online, (available at biosoda.expasy.org), in the Section "Information on Homologous

Gene queries", making it easier for readers to try them out directly without requiring copy-pasting the SPARQL queries from the text of the article.

- *In the last example, the URIs of the same UniProt entry are appeared repeatedly. Can't you use a common variable to specify the query URI only once?*

Authors: We have now replaced the repeated entry with a variable ?uniprot_entry. We also updated the Jupyter notebook accordingly.

Competing Interests: No competing interests were disclosed.

Reviewer Report 27 November 2019

https://doi.org/10.5256/f1000research.23139.r55910

?

Andra Waagmeester

Micelio, Antwerp, Belgium

The authors describe how RDF and SPARQL can be instrumental in combining various resources with knowledge on homologs. The authors take the readers by the hand through some well thought examples of SPARQL queries. They have selected four resources. OMA, OrthoDB, EBI and MBGD. For each resource SPARQL queries are given.
The paper is well written and a welcome contribution to the field. However, there are some issues:

The paper could be improved with a section on applicable ontologies. This maybe as a subsection under materials, where the authors list the resources. Similarly, the used underlying ontologies/controlled vocabularies could be addressed. They are implicitly described in the data model sections, but since this is paper is also presented as a guideline on how to apply SPARQL, it would help the reader to understand how IRIs which are core to the semantic web are selected. Maybe also describe which ontologies in general apply to the narrative of this paper. See Malone *et al*[1].

On various places in the paper the authors argue that SPARQL has a steeping learning curve. I disagree, SPARQL is not different from any other query language. I often argue the contrary, i.e. RDF and SPARQL are rather simple to learn. It is nothing more than wrapping your data in triples and select the proper iri's to represent the knowledge. It can't get any simpler. However, identifying the applicable IRIs and finding a balance between creating IRIs and reusing IRIs makes the process difficult.

"Software availability". I would not call a SPARQL endpoint "software". I suggests that the mentioned SPARQL endpoints can be downloaded and installed locally. Maybe change to "online data access"?

The paper would benefit from a software section, though. If possible it would be nice to show some tool integration. Similar to work by Putman T. *et.al.*[2] where linked data (through Wikidata) is rendered in a webapp.

"If any results are presented, are all the source data underlying the results available to ensure full reproducibility?"

I answered partly, because the SPARQL example queries do not always get results. The SPARQL query ran on the SPARQL endpoint of the EBI did not get results (The query under "a) . *Retrieving EBI pairwise orthologs").* Would it be possible to add captions and referencable numbers to the queries. The authors suggest to copy and paste the query to the applicable sparql endpoint. This might lead to coding issues, due different keyboard layouts that exist, worldwide. I would recommend using clickable links that lead to either a query (e.g. https://w.wiki/CrW) or its results (e.g. https://w.wiki/CrX).

To deal with this (temporarily?) unavailability of the SPARQL endpoint, I would recommend storing the underlying data as rdf in the github repository which also contains the jupyter notebook mentioned. Or at least a subset big enough to demonstrate the results.

It is than possible to reproduce the results using for example the following python gist

```
======
from rdflib import Graph
localGraph = Graph()
localGraph.parse(

query = """ SELECT * WHERE { ....
            ....}
        """
localGraph.query(query)
etc
======
```

References

1. Malone J, Stevens R, Jupp S, Hancocks T, et al.: Ten Simple Rules for Selecting a Bio-ontology. *PLOS Computational Biology*. 2016; **12** (2). Publisher Full Text
2. Putman T, Hybiske K, Jow D, Afrasiabi C, et al.: ChlamBase: a curated model organism database for the Chlamydia research community. *Database*. 2019; **2019**. Publisher Full Text

Is the rationale for developing the new method (or application) clearly explained?
Yes

Is the description of the method technically sound?
Partly

Are sufficient details provided to allow replication of the method development and its use by others?
Partly

If any results are presented, are all the source data underlying the results available to ensure full reproducibility?
Partly

Are the conclusions about the method and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Semantic web, Bioinformatics, data modelling, ontology development.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 08 Jul 2020

Tarcísio Mendes, University of Lausanne, Lausanne, Switzerland

Dear Andra Waagmeester,

We would like to thank you for the insightful comments and suggestions for improvements. We have revised the article in order to take all comments into account. We thus hope that this new version takes into account the remarks and answers the questions raised.

In summary, we have made the following changes:

- We have added all the queries presented in the article to the BioQuery3 interface online, (available at biosoda.expasy.org), Section "Information on Homologous Gene queries", making it easier for readers to try them out directly without requiring copy-pasting the SPARQL queries from the text of the article. Furthermore, we have also created clickable links that point users directly to the corresponding query results in the relevant SPARQL endpoint webpage, in the form of purl.org/orthology/q0 to purl.org/orthology/q9, where the ids Q0-Q9 correspond to the queries presented in the “Protocols” section.
- We have added a subsection "Applicable Ontologies", in the "Materials" section, explaining the ontologies relevant to the orthology domain, used by the four data sources described in the article (OMA, OrthoDB, MBGD and EBI).
- We have improved the explanations regarding the choice of relevant IRIs for formulating SPARQL queries targeting the orthology domain, while clarifying the specific challenges of querying orthology data with SPARQL.
- We have improved the OrthoDB data model figure with small corrections.
- We have included a subsection about “retrieving OrthoDB pairwise orthologs”.

The following paragraphs list the issues identified by you, along with our corresponding answers. Below we provide detailed answers to your comments.

The authors describe how RDF and SPARQL can be instrumental in combining various resources with knowledge on homologs. The authors take the readers by the hand through some well thought examples of SPARQL queries. They have selected four resources. OMA, OrthoDB, EBI and MBGD. For each resource SPARQL queries are given.
The paper is well written and a welcome contribution to the field. However, there are some issues:

- *The paper could be improved with a section on applicable ontologies. This maybe as a subsection under materials, where the authors list the resources. Similarly, the used underlying ontologies/controlled vocabularies could be addressed. They are implicitly described in the data model sections, but since this is paper is also presented as a guideline on how to apply SPARQL, it would help the reader to understand how IRIs which are core to* *the semantic web are selected. Maybe also describe which ontologies in general apply to the narrative of this paper. See Malone et al1.*

Authors: We have added the subsection "Applicable Ontologies" in the "Materials" section. We explained the ontologies relevant to the orthology domain used by the four data sources described in the article (OMA, OrthoDB, MBGD and EBI). In particular, we described the Orthology Ontology (ORTH), but also several other controlled vocabularies used across the four data sources, such as GO annotations, the NCBI taxonomy etc.

- *On various places in the paper the authors argue that SPARQL has a steeping learning curve. I disagree, SPARQL is not different from any other query language. I often argue the contrary, i.e. RDF and SPARQL are rather simple to learn. It is nothing more than wrapping your data in triples and select the proper iri's to represent the knowledge. It can't get any simpler. However, identifying the applicable IRIs and finding a balance between creating IRIs and reusing IRIs makes the process difficult.*

Authors: We have rephrased the abstract to better point out the specific challenges of querying evolutionary data with SPARQL, in particular, formulating property paths to traverse the hierarchical orthologous groups (HOGs). We have also added the new subsections "Applicable Ontologies" and "Choosing the relevant target gene identifiers in RDF" with improved explanations to identify possible relevant IRIs (for example, based on the applied vocabularies) to compose SPARQL queries in the context of the orthology domain. These queries use UniProt accession numbers as a starting point, and use cross-references available in each of the four data sources (OMA, OrthoDB, MBGD and EBI), to retrieve relevant data for target genes of interest.

- *"Software availability". I would not call a SPARQL endpoint "software". I suggests that the mentioned SPARQL endpoints can be downloaded and installed locally. Maybe change to "online data access"? The paper would benefit from a software section, though. If possible it would be nice to show some tool integration. Similar to work by Putman T. et.al.2 where linked data (through Wikidata) is rendered in a webapp.*

Authors: We have renamed the previous "Software availability" section on SPARQL endpoints to “Online Data Access” as suggested. We also added an entirely new section called "Software and Online Data Access", where we describe the integration of the queries in the BioQuery interface available online at biosoda.expasy.org.

- *"If any results are presented, are all the source data underlying the results available to ensure full reproducibility?"I answered partly, because the SPARQL example queries do not always get results. The SPARQL query ran on the SPARQL endpoint of the EBI did not get results (The query under "a) . Retrieving EBI pairwise orthologs"). Would it be possible to add captions and referencable numbers to the queries. The authors suggest to copy and paste the query to the applicable sparql endpoint. This might lead to coding issues, due different keyboard layouts that exist, worldwide. I would recommend using clickable links that lead to either a query (e.g. https://w.wiki/CrW) or its results (e.g. https://w.wiki/CrX).*

Authors: We have created clickable links accordingly. These links point readers directly to the corresponding query results in the relevant SPARQL endpoint webpage. The links are in the form of purl.org/orthology/q0 to purl.org/orthology/q9 where the suffixes q0-q9 after the namespace “ purl.org/orthology/” correspond to the Q0-Q9 queries presented in the “Protocols” section (and are clearly marked in the article as such).

- *To deal with this (temporarily?) unavailability of the SPARQL endpoint, I would recommend storing the underlying data as rdf in the github repository which also contains the jupyter notebook mentioned. Or at least a subset big enough to demonstrate the results.*

Authors: Given that copying the data fragments to our github repository might lead to several issues due to licensing problems, as well as to data size, we have made available a preview of the results directly in the Jupyter notebook. We have included this explanation in the article. We have also added a clarification regarding the availability of the SPARQL endpoints and the relevant support addresses to contact (the support provided by each database owner) in case of long unavailability.

Competing Interests: No competing interests were disclosed.