BIOINFORMATICS ORIGINAL PAPER

Vol. 27 no. 11 2011, pages 1562–1568
doi:10.1093/bioinformatics/btr164

Databases and ontologies

Advance Access publication April 5, 2011

Reasoning with bio-ontologies: using relational closure rules to enable practical querying

Ward Blondé[1,*], Vladimir Mironov[2], Aravind Venkatesan[2], Erick Antezana[2], Bernard De Baets[1] and Martin Kuiper[2]

[1]Department of Applied Mathematics, Biometrics and Process Control, Ghent University, Coupure links 653, 9000 Gent, Belgium and [2]Department of Biology, Norwegian University of Science and Technology, Trondheim, Norway

Associate Editor: Martin Bishop

ABSTRACT

Motivation: Ontologies have become indispensable in the Life Sciences for managing large amounts of knowledge. The use of logics in ontologies ranges from sound modelling to practical querying of that knowledge, thus adding a considerable value. We conceive reasoning on bio-ontologies as a semi-automated process in three steps: (i) defining a logic-based representation language; (ii) building a consistent ontology using that language; and (iii) exploiting the ontology through querying.

Results: Here, we report on how we have implemented this approach to reasoning on the OBO Foundry ontologies within BioGateway, a biological Resource Description Framework knowledge base. By separating the three steps in a manual curation effort on Metarel, a vocabulary that specifies relation semantics, we were able to apply reasoning on a large scale. Starting from an initial 401 million triples, we inferred about 158 million knowledge statements that allow for a myriad of prospective queries, potentially leading to new hypotheses about for instance gene products, processes, interactions or diseases.

Availability: SPARUL code, a query end point and curated relation types in OBO Format, RDF and OWL 2 DL are freely available at http://www.semantic-systems-biology.org/metarel.

Contact: ward.blonde@ugent.be

Supplementary Information: Supplementary data are available at *Bioinformatics* online.

Received on November 29, 2010; revised on March 25, 2011; accepted on March 28, 2011

1 INTRODUCTION

Life Sciences researchers become more and more acquainted with ontologies that support the management of knowledge in their research domains. Many initiatives on biomedical knowledge management have evolved into large Knowledge Bases (KB). Some of these consist of an ontology with a rich semantical content, like the Foundational Model of Anatomy (FMA) (Rosse *et al.*, 2003)—supporting anatomical aspects, SNOMED CT (Truran *et al.*, 2010)—for medical and clinical terms, the Gene Ontology (GO) (Ashburner *et al.*, 2000)—containing cellular information for gene description and the NCBI Taxonomy (Sayers *et al.*, 2010)—holding a classification of living organisms. Other KBs hold a large body of similarly formatted knowledge, like UniProt (UniProt Consortium, 2010)—collecting valuable information about proteins and the Gene Ontology Annotations (GOA) (Barrell *et al.*, 2009)—annotating gene products using the cellular information in GO. Public ontology repositories such as the BioPortal at NCBO (Noy *et al.*, 2009), the Ontology Lookup Service (OLS) (Cote *et al.*, 2008) and BioGateway (Antezana *et al.*, 2009a) make ontologies better accessible for scientists through visualizations, browse menus and search facilities.

Biologists are beginning to accept formal languages and ontologies as instruments to reach consensus while modelling the knowledge of their interest (Antezana *et al.*, 2009c). The large amounts of data that are generated with high-throughput methods call for such a framework (Taylor *et al.*, 2008). In general, a sound framework consists of a common syntax (the symbols and language constructs used), a common semantics (the meaning of the symbols) and common modelling practices (describing how to use the language). Ontology, a domain in philosophy that has long been trying to describe reality, often with the use of logics, is strongly stimulated by its fusion with computer science. Theories that have been developed for over 2000 years can now be applied in automated systems (Petrie, 2009). Since the last decade of the previous century, Description Logics (DL) have been developed as decidable fragments of first-order logic, and some of them with the purpose of efficient reasoning (Baader *et al.*, 2003).

Another important evolution in computer science with respect to ontologies is the emergence of the Semantic Web and the use of Linked Data (Shadbolt *et al.*, 2006). The Semantic Web is viewed as a stack of languages and technologies that make knowledge and data on the internet computer intelligible. This stands in stark contrast to the current World Wide Web, which consists of human readable web sites on the internet. The bottom layer of the Semantic Web stack consists of Internationalized Resource Identifiers (IRI, available at http://www.w3.org/2004/11/uri-iri-pressrelease) that can identify anything, like for instance biomedical concepts. On top of IRIs, there is a layer dealing with the syntax, called XML (Extensible Markup Language, available at http://www.w3.org/XML/), in turn followed by a layer of RDF (Resource Description Framework, available at http://www.w3.org/RDF/), which is useful for querying and inferring graph-based representations (the linked data). Most of the KBs mentioned above are also provided in RDF. The top layer consists of the Web Ontology Language (OWL, available at http://www.w3.org/TR/owl-features/), used for expressing the meaning of knowledge and data (Horrocks, 2003). In October 2009, OWL upgraded to OWL 2, which distinguishes several DL-based

[*]To whom correspondence should be addressed.

sublanguages (profiles), like OWL 2 DL, OWL 2 EL and OWL 2 RL (OWL 2 Profiles, available at http://www.w3.org/TR/owl2-profiles/). By their RDF/XML syntax and by using IRIs, OWL ontologies are computer-manageable, syntactically sound and they provide an unambiguous meaning to well-identified concepts. All these technologies are the most important current standards at our disposal for the implementation of computer-readable ontologies.

Bio-ontologies are meant to be accessible to humans. The investment that biologists put in ontology development is driven by the need to build clear and sound models from the knowledge they have conceptualized collectively. Ontologists have the task to coordinate these efforts into a useful and manageable integrated framework. A consortium that has taken up this challenge is the OBO Foundry, fostering the Open Biomedical Ontologies (OBO) (Smith *et al*., 2007), which should all follow a set of 10 design principles (available at http://www.obofoundry.org/crit.shtml). So far six OBO ontologies (GO, CHEBI, PATO, PRO, XAO and ZFA) have been adopted by the OBO Foundry, while the others remain candidates under review. Many of the OBO ontologies that were developed in the more human-readable OBO Format have been translated into OWL. The Basic Formal Ontology (BFO) (Grenon *et al*., 2004) is developed as an upper level ontology that can integrate all the OBO ontologies. These scientific initiatives —OBO and BFO— are involved in the development of common modelling practices for ontologies across different scientific communities.

Query systems make bio-ontologies accessible to humans. OWL reasoners sustain such a query system on the Semantic Web, but they are very slow and demand too much memory to operate on large KBs, if they work at all. The SPARQL query language (SPARQL, available at http://www.w3.org/TR/rdf-sparql-query/) for RDF performs much better. RDF is perfectly suited for connecting large amounts of knowledge; however, it was not engineered for reasoning purposes.

With the construction of BioGateway (Antezana *et al*., 2009a), we have shown that biomedical resources (OBO ontologies, GOA annotations, UniProtKB/Swiss-Prot and the NCBI Taxonomy) can be interconnected in a single RDF store on the basis of common IRIs, and queried with SPARQL. BioGateway was given some minimal reasoning support for queries through Perl-operated inferences for transitivity of the *is_a* and *part_of* relation types in the OBO ontologies. In Blondé *et al*. (2009), we presented Metarel, a controlled vocabulary for the semantics of relations in RDF, that is very well suited to create inference rules in conjunction with the RDF update language SPARQL/Update (SPARQL 1.1 Update, available at http://www.w3.org/TR/sparql11-update/). Metarel can provide a meaning to a relation between classes as a knowledge statement that takes the basic triple form subject–relation–object. It can be used by simply loading *metarel.rdf*, a meta-ontology for relations, together with KB-derived graphs in a single RDF store.

In this article, we show that semi-automated reasoning on bio-ontologies is possible for a set of closure rules in RDF, with the use of Metarel and SPARQL/Update. We augmented the query system behind BioGateway with inferences from these closure rules, thus further integrating the biomedical resources incorporated in BioGateway. Fully automated OWL reasoning, even on a single OBO ontology, is currently found challenging [e.g. the Sequence Ontology (Holford *et al*., 2010)] or even vexing [e.g. the Cell Cycle Ontology (Antezana *et al*., 2009b)]. By using Metarel as a representation framework for the logical reasoning, we were able to keep an RDF representation that is uncomplicated on both the syntactic level and on the semantic level.

2 DESCRIPTION LOGICS IN THREE STEPS

Description Logics research has kept the hope alive for realizing fully automated reasoning on bio-ontologies. This research promises that any ontology that is modelled in a DL language makes unambiguous sense and that an automated reasoner can answer any logical question about the ontology correctly. However, applying the fully fledged reasoning approach on large, integrated bio-ontologies has proven to be overambitious for two main reasons. First of all, the developers of bio-ontologies, often more experienced in biology than computer science, do not succeed to model all the available knowledge into the rigid language constructs of logics. Consequently, bio-ontologies are full of glitches concerning their logic-based rules (Good and Wilkinson, 2006). Secondly, even if a large bio-ontology succeeds to pass the computational proof of consistency, current automated reasoners are not fast enough for answering queries. Although computer performance continues to increase, the amount of knowledge and data in bioinformatics has been growing even faster. Another hurdle is that being computationally consistent gives us no guarantee that the ontology is actually meaningful and correct.

Even an ontology with imperfections can be useful by providing sensible answers to many real-life questions. In order to better exploit the available ontologies, we need an approach that benefits from DL as much as possible, without insisting on the exclusive use of DL at all stages in constructing a practical query system.

We approach the enabling of large-scale reasoning in three steps: (i) define a logic-based representation language; (ii) build a consistent ontology; and (iii) create inferences for enabling queries. DL reasoning is very useful in the first two steps and has proven already useful for consistency checking of smaller units. However, it is still problematic to implement DL in a query system on a very large scale.

We accomplished the third step for the ontologies in BioGateway by capitalizing on the prior work (by others and ourselves) with respect to the steps (i) and (ii). We minimally adjusted this prior work by a manual curation effort that was restricted to the relation types (types of relations like *is part of*, *is located in*) that were used. This curation effort implies a certain feedback from the last step to the previous steps. Certain language constructs, like defined classes, domains and ranges or number restrictions on relations, may turn out to be very expensive in terms of query time. Alternatively, a relation type used in a given ontology may turn out to create masses of useless inferences. By using Metarel as a semantic framework, and SPARQL/Update as inference tool, we had ample flexibility to engage in a trial and error process to create only those inferences that were useful and necessary. This is a practical alternative to the ambitious approach of DL to execute reasoning as a one-step process without any flexibility for optimization or feedback.

3 REASONING ON BIO-ONTOLOGIES

3.1 All-some relations between classes

Biological knowledge consists almost always of relations between classes (groups) of different individual biological entities. When we express knowledge about cells, proteins or organisms, for instance, we are not referring to a single cell that we observe under a microscope, or a particular mouse that was injected yesterday. We rather refer to classes of many entities that behave in similar ways, and these classes are what we name and identify. In comparison, for example the geographical knowledge domain is strikingly different, as the Atlantic Ocean, New York and Bermuda are large and significant enough to be referred to as individuals with a (usually capitalized) proper name and a proper identifier.

In queries about biological knowledge, we need a logical semantics for relations between classes. The all-some interpretation is the most prominent example to illustrate this (Smith *et al*., 2005). When we relate the classes ‘p53-protein’ and ‘tumour suppression’ with the *has function* relation, it has to mean that all p53-proteins have some tumour suppression as function. This way of using relations provides a very powerful system to infer sound statements of biological knowledge.

Let us give an example using two statements that have the all-some interpretation: ‘every p53-protein *is* some protein’ and ‘every protein *is encoded by* some gene’. From these two, we can derive logically that ‘every p53-protein *is encoded by* some gene’. The inferred statement is sound and it may be the basis for further conclusions.

All the millions of biological classes and the relations between them can be represented as a large network or graph. Queries can be constructed by defining a pattern or subgraph that must match one or several segments of the larger network. Imagine we want to find all the objects that are encoded by a gene and that have some tumour suppression function. Then the search pattern will consist of two triples and one subject that we are interested in: ‘my subject *is encoded by* gene’ and ‘my subject *has function* tumour suppression’. This pattern should match sections of the network, with the middle part of the triples fitting to the relations and the binding elements to the biological classes. The subject ‘p53-protein’ is a possible answer to the query.

We want to use this example to demonstrate the importance of reasoning. What happens if nobody bothered to add ‘every p53-protein *is encoded by* some gene’ explicitly? This absence would prohibit finding ‘p53-protein’ among the list of answers, although this statement follows logically from the two statements described above. It appears that in a good knowledge system all sound statements that are implicit should be made explicit by logical inference, thus augmenting the explicit knowledge in the system by pre-computing. A complete inference of implicit knowledge can be referred to as a ‘closure’.

3.2 Five closure rules for inferring all-some relations

We propose five closure rules for inferring knowledge statements concerning relations between biological classes with an all-some interpretation. These five rules together provide the foundation for the reasoning in step (3) on the current state-of-the-art OBO ontologies and on annotations with OBO ontologies. Annotations of biological subjects imply that an ontology relation and an ontology term are used in the second and third parts of a knowledge statement that is represented as a triple.

Let A, B and C be classes and R, S and T be relation types. For instance, with A = ‘p53-protein’, R = ‘*is encoded by*’ and B = ‘gene’, the knowledge statement $A\ R\ B$ means ‘Every p53-protein *is encoded by* some gene’.

(1) *Reflexivity*: a reflexive closure infers the knowledge statements $A\ R\ A$, where R is a reflexive relation type. For instance, ‘every body *is part of* some body’.

(2) *Transitivity*: a transitive closure infers the knowledge statements $A\ R\ C$, when the knowledge statements $A\ R\ B$ and $B\ R\ C$ exist and R is a transitive relation type. For instance, ‘every kidney *is located in* some body’ follows from ‘every kidney *is located in* some abdomen’ and ‘every abdomen *is located in* some body’.

(3) *Priority over subsumption*: the priority over subsumption infers the knowledge statement $A\ R\ C$, if A is a subclass of B and the knowledge statement $B\ R\ C$ exists, or if the knowledge statement $A\ R\ B$ exists and B is a subclass of C. For instance, ‘every API5-protein *regulates* some cell death’ follows from ‘every API5-protein *regulates* some apoptosis’ and ‘every apoptosis *is* some cell death’.

(4) *Super-relations*: a knowledge statement $A\ S\ B$ is inferred if S is a super-relation of R and the knowledge statement $A\ R\ B$ exists. For instance, ‘every API5-protein *regulates* some apoptosis’ follows from ‘every API5-protein *negatively regulates* some apoptosis’.

(5) *Chains*: a knowledge statement $A\ R\ C$ is inferred if the knowledge statements $A\ S\ B$ and $B\ T\ C$ exist and R holds over a chain of S and T. The relation types R, S and T do not need to be all different. For instance, ‘every API5-protein *negatively regulates* some apoptosis’ follows from ‘every API5-protein *participates in* some anti-apoptosis’ and ‘every anti-apoptosis *negatively regulates* some apoptosis’.

Such closure rules for all-some relations between classes follow directly from rules expressed for (chains of) relations between instances, which are common for DL and OWL. Indeed, if all instances from class A are related to some instances of class B and all instances of class B are related to some instances of class C, then all instances of class A are connected by a chain of two instance relations to some instances of class C. The language features corresponding to the closure rules within step (3) (DL: chains as role constructors; global reflexivity for atomic roles, transitivity for atomic roles and role inclusions as role axioms; existential restrictions of atomic concepts by a role as concept constructors; and concept inclusions with atomic concepts on the left-hand side as terminological axioms) are a subset of those in OWL 2 EL and OWL 2 DL, which warrants efficient reasoning in a decidable semantics.

The semantics of OBO implies additional rules for inferring new knowledge statements. A prominent asset is the use of classes that are logically defined from primitive classes (DL: atomic concepts) through necessary and sufficient conditions. Such defined classes are used in most DL languages, like OWL DL, OWL EL and OWL RL. OBO ontologies have mostly primitive classes with natural language definitions, although logical definitions through intersections of classes are also used. However, the rules in step (3) treat an all-some relation between classes only as a necessary condition, which is not enough for a logical definition.

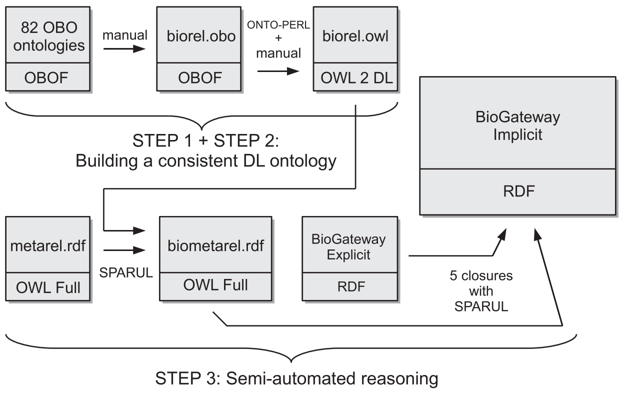

Fig. 1. A practical implementation of the three-step process for reasoning with bio-ontologies through management of relation semantics. A consistent, validated *biorel.owl* in OWL 2 DL contains all the relation types. It is the starting point for applying five important closure rules with a basic RDF tool like SPARQL/Update (SPARUL).

Other features in OBO that do not appear in step (3) are domains, ranges, symmetry, union, disjointness and functionality of relation types, and union and disjointness of classes. It is inherent to the idea introduced above (separating reasoning in three steps) that rules for these features were applied already in step (2) (building a consistent ontology).

Step (3) uses language features that can express knowledge more compactly (DL: logic entailment) and avoids the reasoning problems associated with consistency checking for logically defined classes (DL: satisfiability). If, for instance, the relation type *precedes* was given the range *process* and some annotator or ontology engineer erroneously creates a *precedes* relation to a logically defined class that is disjoint from *process*, then a reasoner should detect this problem in step (2).

An issue not mentioned here is the treatment of individuals (DL: assertional axioms), because they are currently not used in OBO ontologies, nor in the biomedical KBs that are annotated with OBO ontology classes. The individual geographical entities present in OBO's environmental ontology Gazetteer are modelled as singleton classes. In order to model and treat them as individuals, the five rules would need to be complemented with some extra rules. Inverses of relation types, which do not have logical consequences on all-some relations between classes, might also be of use in this extension.

4 METHODS

The large-scale inference of biological knowledge statements was achieved with RDF tools, operating on a merger of Metarel and BioGateway. The merging was relatively straightforward, as the ontologies in BioGateway consisted of the simple triple form subject–relation–object. We curated all the relation types that were used in the OBO ontologies, both candidates and adopted ones, assembling them in a relation ontology called *biorel.obo*. Subsequently, we translated *biorel.obo* into OWL 2 DL and merged it as an RDF graph with *metarel.rdf*. This resulted in the relation graph *biometarel.rdf* for use in BioGateway. Finally, we inferred new knowledge statements as RDF triples by running SPARQL/Update queries over both *biometarel.rdf* and the existing RDF graphs in BioGateway, thereby executing the above-described reasoning approach.

4.1 Manual curation of the relation types

Most relation types in BioGateway originate from the OBO ontologies. All OBO ontologies exist in BioGateway as RDF graphs, providing the opportunity to transform the relational information available in BioGateway with RDF tools. However, standard RDF conversion tools do not properly translate all information embedded in OBO ontologies to RDF, so the work was initiated with the original OBO files for a more expressive translation. All Typedef sections for relation types were separated from all OBO files and through a process of manual curation this long list was reduced to a single valid, consistent OBO file. Text sorting operations and spreadsheets were used to compare and select the best annotated and authoritative relation type entries among the duplicates. In this manner, 833 relation type entries were reduced in a consistent, single-person effort to 365 unique, curated relation types. The resulting OBO file, *biorel.obo*, is available for download at http://www.semantic-systems-biology.org/metarel/biorel.

The most crucial step in our curation process, central to the decision of using the Metarel/RDF framework, was to make a consistent interpretation of the relation types as either object properties (relation types between instances) or all-some relation types between classes. Every relation type used between two terms in an OBO file was interpreted as a metarel:AllSomeClassRelationType and the corresponding relation type in the Typedef section as an owl:ObjectProperty. Relation types that were annotated as 'metadata-tags' in the Typedef section were always interpreted as an owl:AnnotationProperty. This interpretation is entirely consistent with the current standardized practices of translation between OBO and OWL DL (OboInOwl) (Aitken *et al.*, 2008), but it is not consistent with the original interpretation that is still commonly held by many OBO ontology developers. Judging from definitions and tags in OBO's Typedef section, the relation types are most often still viewed as class relation types.

As a consequence, some tags that were introduced in an *ad hoc* manner to solve this ambiguity, like 'inverse_of_at_instance_level' and 'instance_level_is_transitive', were replaced by the standard variants that are captured better by OBO translation tools.

Six relation types, like *has taxonomic rank*, *is valid for taxon* and *is extinct*, have OBO's metadata-tag because they cannot be given an interpretation as object properties. We added this tag to *is integral part of* for the same reason. Its semantics is clear as it can always be written as a combination of the two all-some relation types *is part of* and *has part* in opposite directions: if '*A is integral part of B*', then every *A is part of* some *B* and every *B has part* some *A*. It is interpreted as a *metarel:InvertibleRelationType*, which enables some additional closure rules. However, it does not fit in the general system and it needs to be translated to an annotation property for validation in OWL 2 DL.

A consistent naming system was created, by giving every relation type a name that contains a verb in the third person singular. For instance, the name *after* was replaced by *exists after*, as this seemed to be the intended meaning.

Rules that were only poorly formulated as informal comments were upgraded to sound logic. For instance, the comment that any *starts at end of* implies an *is preceded by* was easily translated to OBO's logic by modelling the former as a subproperty of the latter. The new OBO tag 'holds_over_chain' for creating property chains was exploited to its fullest extent and it was added in several cases. For instance, *is directly preceded by* holds over a chain of *has start* and *is end of*. The 'transitive_over' tag became superfluous through the use of 'holds_over_chain'.

One informally asserted rule stated: 'Gt influences P & Gt variant_of G => G influences P'. This is a chain rule with one object property in the inverse direction. Interpreting the object properties as all-some relation types between classes, we will have *is variant of* from the some-side to the all-side, which does not result in a sound rule on the class level. Indeed, as every Gt is a variant of some entity, it would follow that every entity (everything) influences some P. Implementing such a rule for classes would corrupt the whole knowledge base. The rule was translated to a formal chain rule by using an inverse relation. As no inverse was tagged for *is variant* *of*, the following choice was made: *is influenced by* holds over a chain of *is influenced by* and *is variant of*. This chain goes from the all-side to the some-side and it retains the intended semantics.

Transitivity was added for all the object properties that were tagged as the inverse of a transitive object property. For instance, *is preceded by* was provided with transitivity by the transitivity of *precedes*.

Apart from the names, some dozens of OBO tags for the semantics of relation types had to be altered. Contradictions were nowhere found and the intended semantics could always be retrieved by informal comments and by the way the relation types were actually used in the ontologies.

4.2 Translation to the Semantic Web

The use of the available Semantic Web tools for inferring and querying requires a translation to a Semantic Web language. The current standards are OWL and RDF. BioGateway, an RDF store, does not contain any of the OWL profiles. By using Metarel/RDF as a target framework for the translation, we are, however, still using the standards, because Metarel is valid OWL Full and apart from using class relation types, Metarel is fully compatible with the language constructs used in OWL. Moreover, Metarel being valid OWL Full is technically equivalent with it being valid RDF (http://www.w3.org/TR/owl-ref/). Unlike OWL Full, however, Metarel can connect the class relation types that reside in the RDF of BioGateway with the object properties in Biorel.

We translated *biorel.obo* first to *biorel.owl* using ONTO-PERL (Antezana *et al*., 2008) and adjusted the translated file with some manual curation, which resulted in a valid OWL 2 DL ontology file for the relation types. We added also the chains of relation types, a feature novel in OWL 2 that is in the process of being included in the OboInOwl translation standard. In principle, *biorel.owl* should contain all the expressivity for the rules in Section 3.2, even in RDF.

For our purposes, *biorel.owl* was not practically useful yet, because it contains only object properties, while BioGateway contains only class relation types. We uploaded *biorel.owl* into the relation meta-graph *metarel.rdf* alongside the other RDF ontologies in BioGateway. The merged graph is called *biometarel* and it is this graph that is used for the reasoning process. With SPARUL updates in biometarel, we could connect the object properties of biorel with the class relation types of BioGateway and propagate the semantic rules, like transitivity and chains, to the level of classes.

4.3 Inferring new knowledge statements

Each of the five rules that are required for a query system, as discussed in Section 3.2, corresponds to a single SPARUL/Update query type (Fig. 2). These update queries range over biometarel and the ontology graphs in BioGateway. They need to be operated in a recursive loop until there is no new knowledge statement left that can be inferred.

This practice implies that the entailment of the inferred triples is fully materialized on a hard disk and when this is executed on BioGateway with OpenLink Virtuoso 5.0.8, it shows an acceptable performance. It takes ∼20 h to produce 158 million inferred knowledge statements, which is reasonable compared to an uploading time of 5 h for the 401 million original triples. We would like to point out that not all triples are knowledge statements with a meaningful relation type as a connector. Many triples are used for asserting names, synonyms, definitions, textual annotations, literature references, etc. As these triples are often more verbose, we will call them *verbose triples* as opposed to knowledge statements. For the GO, we get the following numbers: 54 718 explicit knowledge statements, 643 384 verbose triples and 2 031 247 newly inferred knowledge statements. This implies a multiplication factor of 38.12 for the number of knowledge statements, but a multiplication factor of only 3.90 for the total number of triples. For the complete BioGateway we have a multiplication factor of only 1.39.

The relatively low multiplication factor and the high percentage of verbose triples clearly show that a full materialization does not pose many extra storage-related problems for bio-ontologies. It makes no sense to start the reasoning process in a temporal memory only after a query is launched. It takes 20 h to generate all the informative knowledge statements, whereas the majority of typical biologically relevant queries take no more than some seconds to produce an answer. A quick response is absolutely required in the knowledge exploration phase that precedes a more systematic investigation of a new hypothesis. Therefore, the materialization of inferred triples is the preferred practice for bio-ontologies.

```
BASE   <http://www.semantic-systems-biology.org/>
PREFIX metarel:<http://www.metarel.org/>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>

INSERT INTO GRAPH <human_disease_tc> {
  ?Class ?RelType ?Class.
}
WHERE {
  GRAPH <human_disease_tc> {
    ?Class rdf:type ?Type.
  }
  GRAPH <biometarel> {
    ?RelType rdf:type metarel:ReflexiveRelationType.
    ?RelType rdf:type metarel:ClassRelationType.
  }
}
```

Fig. 2. The update query for inferring reflexive relations in the human disease graph. A PERL script parameterizes the 5 SPARQL/Update query types with about 2000 graph names. All the inferences from such small graphs are merged later in the large SSB_tc graph through other update queries. Contrary to OWL, Metarel uses reflexivity and relation types that fit directly between classes.

5 RESULTS

We inferred about 158 million knowledge statements through semi-automated reasoning within BioGateway. The inferences are almost always sound within the intuitive system of all-some relations (an exception: plural forms in ‘Every *Mammalia* is some *cellular organisms*’, following from an incompatible system for defining terms in the NCBI Taxonomy) and they can be accessed directly for any term through BioGateway’s most basic lookup query (results for API5 in Supplementary Material). Most of the inferences are rather trivial if they are considered as a single statement; however, their effect becomes clear to those who are querying the knowledge base. Without the inferences, certain queries either simply return only a fraction of the answers potentially available in the knowledge sources, or they require a lot of very specific knowledge on the architecture of the ontologies in the KB to retrieve full results. The reasoning process would need to be done by the query builder and the resulting queries would become huge and slow.

By pre-computing all the inferences, the hardest part of the reasoning process happens only once in a single although substantial computational effort, but the results are stored and are available for all subsequent queries. The query builder can now concentrate solely on the intended meaning of the relation types that are used, instead of reconstructing this meaning by his query. He can query over the explicit knowledge statements as well as over the implicit ones.

Queries on the RDF graphs with inferred knowledge statements are now short and the answers are more informative and complete. Imagine a cancer researcher who investigates the ASPP1-proteins (Apoptosis stimulating of p53-protein 1) and she finds in direct manual annotations from GOA that proteins of this type are located in the nucleus and in the cytoplasm and that they participate in the processes ‘induction of apoptosis’ and ‘negative regulation of cell cycle’. Now she would like to see which other proteins fulfil these conditions within mammals.

This query involves just a pattern of knowledge statements in the triple form:

- my_subject *is located in* cytoplasm
- my_subject *participates in* apoptosis
- my_subject *participates in* negative regulation of cell cycle
- my_subject *has source* mammal

The query will return all the classes of biological entities (proteins in this case) that can, within BioGateway, be inferred to be located in the cytoplasm, to participate in apoptosis, etc., instead of searching only through the knowledge statements that were once annotated explicitly by an ontology engineer or a manual curator. ‘Mammal’ is too generic to be chosen as an annotation for a source species. Also ‘cytoplasm’, ‘apoptosis’ and ‘negative regulation of cell cycle’ have many sublocations and subprocesses that are often chosen for annotations. The query returns 36 types of proteins that actually fulfil all the conditions, but just 29, 29 and 17, respectively, if only the explicit annotations are queried for ‘cytoplasm’, ‘apoptosis’ and ‘negative regulation of cell cycle’. We still get 13 protein types for explicit annotations on all these three conditions, but none for direct annotations on ‘mammal’. The relatively high numbers of explicit annotations are due to the fact that they are abundant and redundant in meaning, though taken together still incomplete.

We investigated the necessity of each of the five closure rules separately by recreating the inferred version of BioGateway five times and omitting one of the rules during each recreation. We detected that many inferences followed from several different closure rules, however, for all five recreations of the implicit KB, some of the inferences were missing. Four specific biological queries in BioGateway illustrate the practical relevance of each of the closure rules:

- *Query 1*: which are all the biological processes in which a given protein (dnaJ in Chlamydophila felis Fe/C-56) is involved, which are all the other proteins that participate in these biological processes and which cellular locations were annotated for these other proteins? (Bio4 in BioGateway)
- *Query 2*: which are the proteins that have both the nucleus and the endoplasmatic reticulum as inferred locations, compared and ordered for all the organisms in the KB? (Bio5 in BioGateway)
- *Query 3*: what are the subparts of liver parenchyma?
- *Query 4*: which are the developmental stages preceding the unfertilized egg stage, and that are themselves preceded by oogenesis stage S6 (the stage during which follicle cell division ceases)?

The SPARQL translations of these queries and the answers to the queries can be found as Supplementary Material to this article. For each query, we counted the number of answers rendered by either the KB with only the explicit knowledge, on the KB with all the additional inferred knowledge and in each of the KBs that lacked one specific type of closure. The results can be viewed in Table 1.

To demonstrate that the additional answers also make biological sense, we will analyse the queries and the corresponding parts of the KB. Query 1 asks for proteins that are involved in the same biological process as a given process. This means that a protein involved in a subprocess is also a good answer. The query asks generically for proteins in the same process and/or the subprocesses, but without the reflexive closure proteins annotated with the exact same process are disregarded (79 answers). Without any closure, we get only the proteins annotated with the subprocesses on the level immediately below the original process, but not subprocesses of subprocesses (2 answers). Query 2 fails to return proteins that are annotated with sublocations of the nucleus and the endoplasmic reticulum when either the priority over *is_a* or the chain closure is omitted. This depends on the particular engineering of the GO. We find almost exclusively the *is_a* relation type below the nucleus and the endoplasmic reticulum, with only nuclear part *is part of* nucleus and endoplasmic reticulum part *is part of* endoplasmic reticulum, for instance ‘germ cell nucleus *is a* nucleus’ and ‘ARC complex *is a* nuclear part’. The priority over *is_a* propagates *is located in* over all these *is_a*’s, but we need a specific chain rule to propagate *is located in* over *is part of*. The priority over *is_a* generates extra answers for annotations with terms like ‘germ cell nucleus’ (616 answers). But as no protein was annotated with nuclear part nor endoplasmic reticulum part, we get only the explicit annotations on nucleus and endoplasmic reticulum if the priority over *is_a* is omitted (593 answers). Only if both the chain closure and the priority over *is_a* are in place, proteins with annotations in the hierarchy below nuclear part and endoplasmic reticulum part are retrieved (738 answers). Query 3 requires the transitive closure of *is part of* for finding the subparts of ‘liver parenchyma’. Without any closures only ‘liver lobule’, ‘portal lobule’ and ‘portal triad’ are retrieved (3 answers), but not the six more specific terms like ‘bile canaliculus’, which are subparts of the liver lobule and the portal triad. Reflexivity acknowledges that a liver parenchyma is also part of itself (4 answers). Query 4 asks for a series of developmental stages. The ontology of developmental stages uses *starts at end of* as a relation type to connect subsequent stages. *Starts at end of* is a subrelation of the transitive relation type *is preceded by*. That is why only answers are found if both the transitive closure and the super-relation closure are implemented (19 answers).

Table 1. The number of answers to queries compared on explicit knowledge (Exp.) and implicit knowledge (Imp.) and on partial closures where reflexivity (R1), transitivity (R2), priority over subsumption (R3), super-relations (R4) and chains (R5) were omitted

	Exp.	Imp.	R1	R2	R3	R4	R5
Query 1	**2**	118	**79**	118	118	118	118
Query 2	**593**	738	738	738	**593**	738	**616**
Query 3	**3**	10	**9**	**4**	10	10	10
Query 4	**0**	19	19	**0**	19	**0**	19

Bold values show incomplete results.

The results show that every query executed in BioGateway that uses any of the 365 relation types in *biorel.obo* benefits from the reasoning process that has created the inferences. The answers to such a query are complete and they correspond to the logical meaning of the relation types as intended by the ontology engineers. This meaning no longer needs to be simulated in the queries.

6 DISCUSSION

Bio-ontologies and the Semantic Web are two important evolutions for knowledge management in the Life Sciences. They provide a logical framework, universal identifiers and tools for the integration of knowledge. However, in order to become really useful for Life Sciences researchers, both pillars need to mature further.

As the amount of biomedical knowledge keeps growing exponentially, the scalability of Semantic Web tools should be a main concern. Slow queries and memory overflows form a real obstacle for the exploitation of KBs. With this work, we have chosen to enable efficient querying with the most basic semantic features, instead of hampering the query system with advanced, fully automated reasoning.

The large and diverse possibilities of querying RDF demands better browsing and visualizing tools to make the technology more accessible to biologists. Some specific tools for browsing BioGateway are under construction (available at http://www.semantic-systems-biology.org/biogateway/sparql-viewer), but parameterizing and reworking the SPARQL code is still the best option for acquiring all the expressivity of the SPARQL query language. However, the direct relations between classes used in BioGateway may help overcome some of the current shortcomings pertaining to reasoning and browsing.

On the side of the development of bio-ontologies, more efforts are required: ontology engineers should reuse other bio-ontologies to avoid duplication, create appropriate relations, provide identifiers, synonyms, definitions and cross-references. The Semantic Web architecture is perfectly suited for exploiting an orthogonal, cross-linked set of bio-ontologies. BioGateway, with logical inferences in place, can be used to identify the glitches in OBO ontologies.

As a future work, we plan to include more data in BioGateway, like biological pathways, and to test the biological usefulness of the inferences with in-depth queries on very specific research questions. For example, the inferences on GO and the NCBI Taxonomy will allow to compare gene functions across species and kingdoms.

7 CONCLUSION

Many different ontology engineers have collaborated in the coordinated development of more than 80 OBO ontology files. We have brought consistency to the stack of relation types in these files by gathering all the relation types in Biorel and translating them to OWL 2 DL. After merging the OWL translation of Biorel with Metarel in the RDF store BioGateway, we could infer 158 million previously hidden knowledge statements from the explicitly asserted knowledge in the OBO ontologies, GOA annotations for about 2000 species, UniProtKB/Swiss-Prot and the NCBI Taxonomy. The inferred knowledge statements can be used for biological hypothesis generation through querying. The success of our methodology is due to the soundness of OBO ontologies, the use of Semantic Web tools and the semi-automated approach of reasoning.

Our work shows that a small set of simple rules for bio-ontologies results in efficient practices for reasoning and querying. As many researchers are involved in building bio-ontologies, more restrictive guidelines and principles for building bio-ontologies are required in order to obtain more uniformity and reach more convergence for knowledge management in the Life Sciences.

ACKNOWLEDGEMENTS

We wish to thank Nirmala Seethappan, Einar Jensen, Kent Overholdt and Bjørn Lindi for their IT support.

Funding: European Science Foundation (ESF) for the activity entitled Frontiers of Functional Genomics (to W.B. and E.A.); FUGE Mid-Norway (to V.M.).

Conflict of Interest: none declared.

REFERENCES

Aitken,S. *et al.* (2008) OBO Explorer: an editor for open biomedical ontologies in OWL. *Bioinformatics*, **24**, 443–444.

Antezana,E. *et al.* (2008) ONTO-PERL: an API for supporting the development and analysis of bio-ontologies. *Bioinformatics*, **24**, 885–887.

Antezana,E. *et al.* (2009a) BioGateway: a semantic systems biology tool for the life sciences. *BMC Bioinformatics*, **10**, S11.

Antezana,E. *et al.* (2009b) The Cell Cycle Ontology: an application ontology for the representation and integrated analysis of the cell cycle process. *Genome Biol.*, **10**, R58.

Antezana,E. *et al.* (2009c) Biological knowledge management: the emerging role of the Semantic Web technologies. *Brief. Bioinform.*, **10**, 392–407.

Ashburner,M. *et al.*; Gene Ontology Consortium (2000) Gene Ontology: tool for the unification of biology. *Nat. Genet.*, **25**, 25–29.

Baader,F. *et al.* (2003) *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, New York, NY, USA.

Barrell,D. *et al.* (2009) The GOA database in 2009-an integrated Gene Ontology Annotation resource. *Nucleic Acids Res.*, **37** (special issue), D396–D403.

Blondé,W. *et al.* (2009) Metarel: an ontology to support the inferencing of Semantic Web relations within Biomedical Ontologies. In *Proceedings of the International Conference on Biomedical Ontologies (ICBO)*, Nature Precedings, Doc. 3562, v.1.

Cote,R. *et al.* (2008) The Ontology Lookup Service: more data and better tools for controlled vocabulary queries. *Nucleic Acids Res.*, **36**, W372–W376.

Good,B. and Wilkinson,D. (2006) The Life Sciences Semantic Web is full of creeps! *Brief. Bioinform.*, **7**, 275–286.

Grenon,P. *et al.* (2004) Biodynamic ontology: applying BFO in the biomedical domain. *Stud. Health Technol. Inform.*, **102**, 20–38.

Holford,M. *et al.* (2010) Using semantic web rules to reason on an ontology of pseudogenes. *Bioinformatics*, **26**, i71–i78.

Horrocks,I., *et al.* (2003) From SHIQ and RDF to OWL: the making of a web ontology language. *J. Web Semant.*, **1**, 7–26.

Noy,N. *et al.* (2009) BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Res.*, **37(S)**, W170–W173.

Petrie,C. (2009) The Semantics of 'Semantics'. *IEEE Internet Comput.*, **13**, 94–96.

Rosse,C. and Mejino,J. (2003) A reference ontology for biomedical informatics: the Foundational Model of Anatomy. *J. Biomed. Inform.*, **36**, 478–500.

Sayers,E. *et al.* (2010) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.*, **38**, D5–D16.

Shadbolt,N. *et al.* (2006) The Semantic Web revisited. *IEEE Intell. Syst.*, **21**, 96–101.

Smith,B. *et al.* (2005) Relations in biomedical ontologies. *Genome Biol.*, **6**, R46.

Smith,B. *et al.* (2007) The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat. Biotechnol.*, **25**, 1251–1255.

Taylor,F. *et al.* (2008) Promoting coherent minimum reporting guidelines for biological and biomedical investigations: the MIBBI project. *Nat. Biotechnol.*, **26**, 889–896.

The UniProt Consortium (2010) The Universal Protein Resource (UniProt) in 2010. *Nucleic Acids Res.*, **38**, D142–D148.

Truran,D. *et al.* (2010) SNOMED CT and its place in health information management practice. *Health. Inf. Manag. J.*, **39**, 37–39.