BioPortal as a Dataset of Linked Biomedical Ontologies and Terminologies in RDF.

Manuel Salvadores, a,∗ Paul R. Alexander, a Mark A. Musen a and Natalya F. Noy a
a Stanford Center for Biomedical Informatics Research
Stanford University, US
E-mail: {manuelso, palexander, musen, noy}@stanford.edu.

Abstract. BioPortal is a community-based repository of biomedical ontologies—the largest such repository, with more than 300 ontologies to date. This set includes ontologies that were developed in OWL, OBO and other formats, as well as a large number of medical terminologies that the US National Library of Medicine distributes in its own proprietary format. We have published the RDF version of all these ontologies at http://sparql.bioontology.org. This dataset contains 190M triples, representing both metadata and content for the 300 ontologies. We use the metadata that the ontology authors provide and simple RDFS reasoning in order to provide dataset users with uniform access to key properties of the ontologies, such as lexical properties for the class names or provenance data. The dataset also contains 9.8M cross-ontology mappings of different types, generated both manually and automatically, which come with their own metadata.

Keywords: biomedical ontologies, BioPortal, RDF, linked data

1. Introduction

In our laboratory, we have developed BioPortal, a community-based ontology repository for biomedical ontologies [16,1]. Users can publish their ontologies to BioPortal, submit new versions, browse the ontologies, and access the ontologies and their components through a set of REST services, SPARQL and dereferenceable URIs.

Over the past four years, as BioPortal grew in popularity, research institutions and corporations have used our REST APIs extensively. The number of unique visitors to BioPortal site has tripled in the last 6 months. In December 2011, we released a public SPARQL endpoint, http://sparql.bioontology.org, to provide direct access to our datasets in RDF. We had numerous requests from users for the SPARQL endpoint, which would enable them to query and analyze the data in much more precise and application-specific ways than our set of REST APIs allowed.

This paper describes the Linked Data aspects of BioPortal’s ecosystem and the structure of our linked datasets in RDF. In addition, we describe the process that we used to transform different ontology formats into RDF and the mappings between ontologies.

2. Biomedical Ontologies in BioPortal

Researchers and practitioners in the Semantic Web normally deal with two types of data: (1) ontologies, vocabularies or TBoxes; and (2) instance data or simply data. It is important to clarify that BioPortal’s content is almost exclusively ontologies and related artifacts. By contrast, most other datasets of the Linked Data Cloud focus on instance data and ontologies and schemas play only a small role there. In the biomedical domain, ontologies play a very active and important role and many ontologies and vocabularies are extremely large, with tens of thousands of classes and complex expressions. For example, SNOMED CT, one of the key terminologies in biomedicine, has almost 400,000 classes [18]. The Gene Ontology (GO) has
34,000 classes [9]. These ontologies and terminologies are updated on a regular basis, some very frequently. For example, a new version of GO is published daily.

2.1. Ontology Formats

There are three main ontology formats in BioPortal:

- The **OBO format** is the format that many developers of biomedical ontologies prefer because of its simplicity. The OBO Editor, an tool that many ontology developers in biomedicine use, produces ontologies in this format. The OWL API now provides a de facto standard translation from OBO Format to OWL 2.

- The **Rich Release Format (RRF)** is primarily used by the US National Library of Medicine to distribute the vocabularies that constitute the Unified Medical Language System (UMLS) [13].

- **OWL** is a W3C recommendation for representing ontologies on the Semantic Web. Unlike OBO and RRF, OWL was not specifically designed for the biomedical domain.

At the time of this writing, BioPortal contains 167 OWL, 110 OBO and 25 RRF ontologies.

3. RDF Dataset Description

There are three main components in the BioPortal dataset: ontology content, metadata and mappings. The following sections describe each of these in detail.

3.1. Ontology Content

The core of the BioPortal dataset is the actual content of each ontology that users submitted to BioPortal. BioPortal, as a repository, keeps multiple versions of each ontology but only exposes the latest version of each. For OBO and OWL ontologies, the content is in the materialized view of the ontology produced by computing the closure of the owl:imports statements [17].

Ontologies in BioPortal vary in their content and structure. There are very rich representations, such as those found in the NCI Thesaurus [10], which has 111K rdfs:subClassOf relations [4]. There are also terminologies, with no single transitive taxonomic relation, such as Medical Subject Headings (MeSH) [3].

The ontology authors use different properties to represent common relations and attributes. For instance, in order to represent the class hierarchy they use rdfs:subClassOf, skos:narrower, obo:is_a, or some other instance of owl:TransitiveProperty. The ontologies in BioPortal use 17 different properties to represent a preferred label of a term, and 28 different properties to store synonyms—even though standards, such as SKOS, provide recommendations for the properties to use in these cases. The ontology authors specify which properties they use for these common annotations as part of the metadata for their ontology. In order to provide the users of the BioPortal dataset with a uniform access to properties such as preferred labels, synonyms, definitions, and so on, we link these different properties to the standard SKOS properties using rdfs:subPropertyOf relation. For example, properties that individual ontologies use for preferred labels all become subproperties of skos:prefLabel in a “globals” graph. As the result, we have a set of common predicates to query over lexical annotations across ontologies. The globals graph contains a hierarchy of properties that maps each custom annotation to one of the standard predicates. We use this hierarchy of predicates to rewrite internally the SPARQL query using backward-chaining reasoning. Figure 1 shows an example of a SPARQL query for an ontology that uses a custom predicate to record preferred labels. In this case, the user does not need to know the specific predicate and she can query on the standard skos:prefLabel.

Hierarchies and lexical annotations are common ground for most of the ontologies and we try to provide capabilities to facilitate querying across them. Ontologies also contain other types of expressions and statements. The variability of predicates makes our data very sparse and one has to focus on small subsets of the ontologies to find common constructs. Our store contains 2,541 different predicates and the occurrence of subjects by predicate has a long-tail distribution (Figure 2). The twenty most popular predicates are used...
in more than $10^6$ statements, accounting for 75\% of the total number of triples. These top 20 predicates are from standard vocabularies, which are used to record hierarchies or lexical annotations, such as rdfs:type, skos:prefLabel, rdfs:label, rdfs:subClassOf, and so on. Figure 2 also shows that the percentile distribution becomes almost flat after the 500 predicate mark; these 500 predicates constitute 98\% of the dataset.

### 3.2. Metadata

In addition to ontology content, we track a set of metadata related to each ontology in the system. We represent the metadata using an OWL ontology that we developed for this purpose, the BioPortal Metadata Ontology [14], which extends the Ontology Metadata Vocabulary (OMV). The metadata is a set of instances in this OWL ontology. The two main entities in the metadata are meta:VirtualOntology and omv:Ontology. bp:VirtualOntology represents a container for all versions of an ontology; an omv:Ontology represents a particular ontology version. Figure 3 describes the connections between these two elements.

Among other properties, BioPortal’s metadata contains properties to record names, descriptions, submission date, author, contact email, project website, etc. Figure 4 shows an RDF/Turtle example containing some relevant predicates.

### 3.3. Mappings

Mappings between terms in different ontologies constitute an important part of the BioPortal repository [15]. Users can submit mappings to BioPortal through the Web interface or the REST APIs. In addition, the BioPortal team runs a series of processes to generate mappings automatically.

A mapping in BioPortal connects two terms from different ontologies. It may also connect one term to many terms (this case is rare, and we do not cover it here). We abstract the mappings into entities that record the provenance information of the mapping: the process that generated the mapping, when and how it was produced, the user who submitted it, the type of relation between classes, etc. This information is represented in two sets of triples (a) the mapping itself and (b) the process information, which is referenced by all the mappings that the process generated (Figure 5).

We use SKOS-based relationships to state the level of similarity between terms. The predicates that we use include skos:exactMatch, skos:closeMatch, or skos:relatedMatch (Table 1).

There are different types of mappings in BioPortal, which currently include the following:
Lexical Mappings (LOOM): These are lexical mappings that we generated by performing simple lexical comparison between preferred labels and preferred labels and alternative labels for terms. There are 6.2M skos:closeMatch mappings of this type.

Xref OBO Mappings: Xref and Dbxref are properties that developers of ontologies in OBO use to refer to an analogous term in another vocabulary. We generated 2.2K based on the Xref properties in OBO ontologies in BioPortal (skos:relatedMatch).

CUI Mappings from UMLS: Similar terms from different vocabularies in UMLS are assigned the same Concept Unique Identifier (CUI). We generated 3.1M skos:closeMatch mappings between the terms in UMLS vocabularies using CUIs as join point. This set of mappings represent the largest human-curated set in BioPortal.

URI-based Mappings: We generated identity mappings between classes in different ontologies that have the same URI. 203K skos:exactMatch mappings fall into this category.

User Submitted Mappings: Visitors to the BioPortal site can create mappings manually. There are 12K mappings submitted in this way.

Other mapping statistics between ontologies can be found at the BioPortal group in thedatahub.org [2].

### Table 1

<table>
<thead>
<tr>
<th>SKOS Predicate</th>
<th>Number of Mappings</th>
</tr>
</thead>
<tbody>
<tr>
<td>skos:closeMatch</td>
<td>9,492,690</td>
</tr>
<tr>
<td>skos:exactMatch</td>
<td>361,495</td>
</tr>
<tr>
<td>skos:relatedMatch</td>
<td>2,255</td>
</tr>
</tbody>
</table>

4. Structure of Named Graphs

In sparql.bioontology.org, we have deployed a multi-graph structure where ontologies and mappings reside in different graphs. We use two graphs for each ontology in the repository: one graph for the ontology content and another for the ontology metadata. The metadata graph ID is a URI equivalent to the virtual URI in BioPortal and the content graph ID is a URI where the last fragment is the acronym of the ontology. For instance, the following two graph names are SNOMED content and metadata respectively:

**Content:**

http://bioportal.bioontology.org/ontologies/SNOMED

**Metadata:**

http://bioportal.bioontology.org/ontologies/1353/metadata

There is an RDF statement that links the metadata graph with the content graph (Figure 3). The SPARQL query in Figure 6, shows how to retrieve all IDs for ontology content graphs.

```sparql
PREFIX meta: <http://bioportal.bioontology.org/metadata/def/>;
SELECT DISTINCT ?version ?graph WHERE {
  ?version meta:hasDataGraph ?graph
}
```

Fig. 6. SPARQL Query to retrieve Content Graph IDs

As we have mentioned earlier (Section 3.1), we materialize the ontology content with its imports into a single graph. Therefore, programs that query only one ontology need to retrieve only the named graph where that ontology is located. This approach results in data redundancy in our store but facilitates query articulation by making a one-to-one relation between ontologies and named graphs.
5. API Keys and Private and Licensed Ontologies

BioPortal implements a data sharing model that allows ontology owners to control who can access their data. Ontology administrators can set a visibility flag, declaring an ontology as public, licensed, or private. If the ontology is public, then all users can access it. If the flag is set to licensed, then users must provide their license for the ontology in order to access it. Private ontologies are accessible only to the users to whom the ontology administrators have specifically granted access. In the REST APIs, we control the access by requiring users to pass an API key that identifies the user in the HTTP request.

We have mimicked this behavior in the SPARQL endpoint. As backend storage we use 4store [11]. Our team has modified 4store’s code base in order to provide access control at the graph level. A user’s API key needs to be included in the SPARQL HTTP call as a parameter and our 4store extensions will process the SPARQL query using only the graphs that the user is allowed to access [5].

6. Linked Data Resources

In addition to SPARQL access, BioPortal provides dereferenceable terms and ontology URIs. Linked Data crawlers can retrieve the entire content of an ontology with one HTTP request directed to the ontology URI. For instance, the Cell Line ontology can be retrieved in RDF with:

```
$ curl -H 'Accept: application/rdf+xml'
   http://purl.bioontology.org/ontology/CLO
```

Individual terms can be resolved in RDF by dereferencing a specific term URI. Term URIs are normally in the name space that ontology authors have defined, which is outside of BioPortal’s domain. To provide linked data for these URIs, our web front-end provides permanent URLs for each ontology term using a PURL server. We configured our PURL server to redirect URLs of the following form:

```
http://purl.bioontology.org/ontology/{ACR}/{SHORT_ID}
```

Our PURL server will redirect this URL to get information about the term with the ID SHORT_ID in the ontology identified by a unique acronym ACR. For example, the following URL uses an ontology acronym NCIt, which refers to NCI Thesaurus, and short id “Haemophilus_influenzae” to access information about this term:

```
http://purl.bioontology.org/ontology/NCIt/Haemophilus_influenzae
```

We use content negotiation to determine whether we should provide the term information in HTML or RDF.

7. RDF Dataset Creation Workflow

In order to support multiple ontology formats, BioPortal currently utilizes two applications, LexEVS and Protégé. LexEVS is responsible for parsing and storing terminologies in formats that are primarily used in the biomedical domain: OBO Format and RRF. Protégé handles ontologies in OWL, OWL2, and Protégé Frames. Unfortunately, Protégé and LexEVS lack some functionality that is generally accepted as standard on quad storage platforms, including the ability to query easily across records in different ontologies and scalability into the tens-of-million triples range.

Prior to our recent quad store implementation, our data had not been stored as triples in our backend systems and therefore we need to follow a different workflow for each format to expose the existing content as RDF triples. Figure 7 shows the pipeline and tools that we used to generate RDF triples from the ontologies.

---

Footnote:

1. https://github.com/ncbo/umls2rdf
To process OBO and OWL ontologies, we use the OWL-API [12]. The OWL-API can read the OBO syntax and all the OWL syntaxes (e.g: OWL/XML, Manchester, RDF and Manchester syntax). We also use the OWL-API to extract the import closure. We fetch imports from the web and materialize them, saving the whole materialized ontology in the data store.

We assert the BioPortal’s metadata in the triple store using the Protégé API.

We generate the mappings between ontologies directly in RDF.

We process the pipeline in Figure 7 daily at midnight PST time. Ontology changes are propagated to the triple store overnight and updates cannot be seen until the next day.

8. Summary

The BioPortal Linked dataset provides uniform access to a widely used repository of more than 300 biomedical ontologies. The dataset contains the ontologies themselves, the metadata about the ontologies, and the mappings between terms in different ontologies. It supports de-referencing about URIs for whole ontologies and individual terms in the ontologies. To reflect the linked open data aspect of BioPortal we have registered the ontologies at the thedatahub.org [2].

Appendix: Other Tools and Resources

The BioPortal project is committed to releasing its code as Open Source. We have developed the following components as part of this work:

- NCBO’s 4store clone: The NCBO project maintains an updated clone of the 4store repository where we implement bugs fixes and add new functionality. This work feeds back into 4store’s main distribution and the code bases are merged regularly.

- The Web front-end at sparql.bioontology.org is a python/django application that integrates SNORQL.js to provide direct SPARQL access via web browsers.

Both the 4store clone and the SPARQL proxy are at the NCBO’s github repository https://github.com/ncbo.

Other resources include links to documentation:
- Code examples in Java, Perl, Python, Javascript and Ruby to programmatically access our SPARQL endpoint [6].
- Project Wiki Documentation with SPARQL information [7].
- Presentation with introduction to RDF and SPARQL and details on how to access our RDF store [8].

Acknowledgments

This work was supported by the National Center for Biomedical Ontology, under grant U54 HG004028 from the National Institutes of Health.

References