BioDSL: A Domain-Specific Language for mapping and dissemination of Biodiversity Data in the LOD

Kleberson J. do A. Serique[1], José L. Campos dos Santos[2], Dilvan A. Moreira[1]

[1]University of São Paulo (USP)
CEP: 13566-590 – São Carlos – SP – Brazil

[2]National Institute for Amazonian Research (INPA)
CEP.: 69060-001 – Manaus – AM – Brazil

junio.serique@gmail.com,laurindo.campos@inpa.gov.br,dilvan@gmail.com

Abstract. *Currently, Linked Open Data (LOD) have enabled integrated data sharing across disciplines over the Web. However, for LOD users, in areas such as biodiversity (which massively use the Web to disseminate data), the task of transforming data file contents in CSV (Comma Separated Value) to RDF (Resource Description Framework) is not trivial. We have developed a new approach to map data files in CSV to RDF format based on a domain-specific language (DSL) called BioDSL. Using it, biodiversity data users can write compact programs to map their data to RDF and link them to the LOD. Biodiversity vocabularies and ontologies, such as Darwin Core and OntoBio, can be used with BioDSL to enrich user data. Existing tools are exclusively focused on mapping (CSV to RDF), offering little or no support for linking data to the LOD (interconnecting user entities to LOD entities). They also are more complex to use than BioDSL.*

1. Introduction

In biology, data are understood as a collection of facts that require interpretation of their meaning in order to become knowledge. This interpretation is performed only by humans, but the interpretation of extreme large data sets is only possible with the use of computers, ideally, with high performance capabilities.

In recent years, biodiversity open data have become openly available online in sites such as GBIF[1]and SpeciesLink[2]. However, these data are available, in most cases, in CSV (Comma Separated Values) and other formats that do not have explicit semantic information about data [Van Der Waal et al. 2014]. While online data aggregators (such as GBIF) have helped in increasing the amount of biodiversity data available in digital format, the meaning of these aggregated data is often ambiguous [Moura et al. 2012].

To tackle this problem, a recent and growing movement, called Linked Open Data (LOD), uses Semantic Web technologies for sharing and linking data over the Web. LOD refers to a set of best practices for publishing data with semantic information so that it can be interlinked, referenced across datasets, and explored by automatic processing[Berners-Lee 2006]. LOD sites use the RDF model to represent knowledge in a simple structure. In RDF, data is represented as statements. Each statement is represented

[1]http://www.gbif.org/
[2]http://splink.cria.org.br/

by a triple (S, P, O), where S is a URI (Uniform Resource Identifier) for the subject, P is a URI for the predicate, and O is either an URI or literal for the object.

Nowadays, LOD sites contain a large number of linked openly available datasets that cover a wide variety of disciplines, including biology. They form the LOD data cloud. Even though there is a large amount of biological datasets in the LOD cloud, no significant amount of biodiversity datasets are available for reuse and sharing. Also, biodiversity is still absent from works related to the development of ontologies and Semantic Web technologies [Walls et al. 2014]. Ontologies are used to provide a common vocabulary for a domain of interest and define the logical relationships that hold between these vocabulary items [Matsubara et al. 2009]. Furthermore, formal ontologies facilitate the discovery of implicit knowledge, through the reasoning process.

Once available on the LOD cloud, biodiversity datasets become more easily accessible by analytical tools, such as scientific workflows. Such tools can be used not only to test and integrate biodiversity data, from different sources, but also to integrate them to sources from other scientific fields (equally important to understand biodiversity phenomena).

To increase the availability of biodiversity data on the LOD cloud, one approach is to convert data, in CSV and other tabular formats used by biodiversity researchers, to RDF. Current tools to map data from CSV format to RDF offer no or little support to link these data to data already available on the LOD cloud.

This paper presents BioDSL, a new approach for the mapping of biodiversity data, in tabular format (CSV), to RDF. BioDSL is part of a bigger project to build a semantic infrastructure for biodiversity data.

The remainder of this paper is structured as follows. Section 2 presents the related work. Section 3 presents the BioDSL language. Section 4 concludes with some suggestions for future work.

2. Related Work

The Semantic Web provides technologies conducive to scientific data integration and dissemination. The first step, to take advantage of such technologies, is to transform tabular data (CSV) into a format readable by machines, such as RDF. The next step is to follow Linked Open Data principles when generating these data[Berners-Lee 2006].

There are two general approaches to map CSV data to RDF, the row and cell-based translations. The row-based translation is the most common. It assumes that each row describes a subject and that each column represents a property[Dimou et al. 2014]. This approach allows us to quickly get RDF from a CSV file. The cell-based translation assumes that each row can represent more than one subject, or, in many cases, that properties may come from a vocabulary or ontology.

Currently, there are several tools to convert structured data, as CSV files, to RDF[3]. In order to find works related to BioDSL, a literature review was performed. Among the works found, we highlight here the ones closely related to our proposal, such as RDF

[3]http://www.w3.org/wiki/ConverterToRdf

Refine[4], the SemantEco Annotator, the csv2rdf4lod[5], the Apache Any23[6], the recommendations of RDB2RDF[7], the RML mapping language and the Sparqlification mapping language (SML)[8]. A more general overview of mapping tools for structured sources, such as CSV, is given in [Unbehauen et al. 2012].

The RDB2RDF working group of the W3C produced two recommendation for the conversion of Relational Database (RDB) and CSV data to RDF [Scharffe et al. 2012]. The first recommendation is the Direct Mapping of Relational Data to RDF[9] that is similar to row-based translation. This approach allows to quickly get RDF from a CSV file, provided you follow the RFC4180[10] standard: the file should provides a header for columns. The second uses the R2RML language[11] to perform the mapping and enable the use of terms from vocabularies and ontologies. R2RML mappings are themselves RDF graphs and are written down in Turtle syntax[Stadler et al. 2015].

The Apache Any23 (Anything To Triples) is a library, a web service and a command line tool that extracts structured data in RDF format from a variety of Web documents (CSV, HTML, Microformats, etc). It is used in a large number of applications, such as Sindice. Apache Any23 converts files in CSV to RDF following an extraction algorithm[12]. This algorithm requires that: i) the CSV files should be compatible with the RFC4180 standard, headers will be used as RDF properties; ii) a base URI should identify the CSV file. However, to perform the interconnection of the CSV file records with other LOD entities, the user must put their entities URIs in a new column in the original CSV file.

The RDF Refine [Maali et al. 2011] is an extension to OpenRefine[13], where the data can be converted to RDF. It requires that users define a skeleton for the RDF as a tree structure, then it creates the mappings between the columns in the table and the nodes of the tree. One of the highlights feature of the RDF Refine is the OpenRefine data reconciliation tool. It allows the user to perform searches for URI entities on the LOD, such as DBpedia[14] to associate them with entities in the CSV file columns [Maali et al. 2011]. However, this procedure requires some adjustments, transposition and additions of columns, thereby altering the structure of the original CSV file.

The SemantEco Annotator is similar to RDF Refine. It provides a simple user interface, keeping all interactions for mapping the RDF separate from the content of the original data table, which remains unchanged [Seyed et al. 2013]. This annotator serves as a front-end to the csv2rdf4lod conversion tool.

The csv2rdf4lod tool [Lebo and Williams 2010] converts tabular data files, such as CSV, to RDF according to the interpretation of coded parameters using a specific vo-

[4]http://refine.deri.ie
[5]https://github.com/timrdf/csv2rdf4lod-automation/wiki
[6]http://any23.apache.org/
[7]http://www.w3.org/2001/sw/rdb2rdf
[8]http://sparqlify.org/wiki/SML
[9]http://www.w3.org/TR/rdb-direct-mapping/
[10]http://www.ietf.org/rfc/rfc4180.txt
[11]http://www.w3.org/TR/r2rml/
[12]http://any23.apache.org/dev-csv-extractor.html
[13]http://openrefine.org
[14]http://dbpedia.org

cabulary for conversion[15]. This tool has two types of conversion: raw conversion and enhanced conversion. The first uses the row-based translation, while the latter needs a RDF file encoding a vocabulary to make the conversion. This file also provides the provenance history of the operations that led from the original CSV file.

SML is an RDB2RDF human readable mapping language with the same expressiveness as R2RML. SML provides virtual RDF graphs over relational databases or CSV files. This language is part of the Sparqlify platform that integrate several components of a Web application in a SPARQL endpoint. SML is based on R2RML. It has equal expressiveness, but it is less verbose than R2RML[Stadler et al. 2015].

The RML mapping language is an extended version of the R2RML language. RML keeps the mapping definitions, as in R2RML, but excludes database-specific references from the core model[Dimou et al. 2014]. The main difference between them are the input sources. In R2RML, they can be a database table or CSV file, while in RML they can be a broad set of input sources that, together, describe a certain domain.

Table 1. Comparison of the most common features of tools mapping CSV to RDF

Feature	RDF Refine	SemantEco	csv2rdf4lod	Apache Any23	RML	SML	BioDSL
Add Ontologies	-	O	X	-	X	X	X
Not Change CSV columns	-	X	X	X	X	X	X
LOD Instance matching	X	-	-	-	-	-	X
Make RDF model	X	X	X	X	X	X	X
Make OWL model	-	-	-	-	-	-	X
Infers OWL model	-	-	-	-	-	-	X
Export to SPARQL endpoint	-	-	-	-	-	X	X
Show mapping inconsitence	-	O	-	-	-	-	X
UI editor interface	X	X	-	-	-	O	O
Row-based translation	X	X	X	X	X	X	X
Cell-based translation	X	X	X	-	X	X	X
Visual-based mapping	X	X	-	-	-	-	O
Code-base mapping	-	-	X	-	X	X	X
CSV Data type verification	X	O	-	-	-	-	X
Reusability of the mappings	-	-	X	-	X	X	X
X Developed O Under Development - No Information							

Table 1 presents a feature list with comparisons between these tools features and BioDSL's. The main drawback of most of these data mapping solutions is the assumption that each row describes a single entity, such as row-based translation. Moreover, existing tools are exclusively focused on mapping the CSV data to the RDF model. They do not attempt to interconnect their source entities with existing entities in the LOD cloud.

Just RDF Refine makes an instance matching to automatically find LOD cloud entities URIs for reuse. This allows linking the data sets with other facts from other

[15]http://purl.org/twc/vocab/conversion

datasets (which allows data enrichment). Also, these tools do not allow that users create an integrated model with all used ontologies and mapping data as an OWL file. That could be used in other ontologies tools, such as Protégé. A missing, but useful (specially for novices), feature in these tools is a SPARQL endpoint. It allows other users to make queries against datasets using a Semantic Web standard. It also allows instance matching to improve the linking between datasets. A much desired feature is to show inconsistent mappings: it can be an ontological inconsistency or a syntax error, in case of script-based mapping. Data type errors must also be shown, thus the users can improve their data with accurate data type values.

To address the remaining gaps in the CSV to RDF conversion process of biodiversity data, BioDSL was developed as a DSL to make the complex process of conversion transparent to end users, covering from data integrity checks to ontological reasoning in OWL. In addition, BioDSL allows mapping of entities, present in CSV data sets, to existing entities in the LOD cloud, thus facilitating the integration of new data with other datasets already in the LOD cloud. It has the potential of enabling the discovery of new knowledge implicit in the data source.

3. The BioDSL Language

The BioDSL implementation uses the Groovy programing language. Groovy has native support for the development of DSLs and is compatible with the Java programing language and many Semantic Web's APIs.

In a typical biodiversity CSV file, each column may represent entities of different types (classes) and each line encode several entities related to each other, such as collected species, collection sites, institutions, collectors. These entities can also be repeated in different lines.

BioDSL has a declarative syntax based on objects and functions. For instance, the CSV file is represented by an object, called csv, whose properties refer to the column names (such as *csv.institutionID*). The main function, Map, does the actual mapping between CSV file and RDF. Other auxiliary functions define how URIs will be created, for instance, they can associate entities, from the CSV file, to LOD entities through their URIs. The following subsections will describes BioDSL syntax.

3.1. CSV file

Figure 1 shows the way to load a CSV file into a BioDSL script. The *addCsv* method, in Figure 1 line 1, receives a path or a URI to load a CSV file. The function *ignoreRow* declares numbers of lines that will be ignored (shown in Figure1 line 2). To skip more than one line, users must use a list of values corresponding to the line numbers (shown in Figure 1 line 3).

3.2. Loading Ontologies and defining Prefixes

Ontologies can be loaded into BioDSL and their vocabularies used to name and classify entities. The syntax to load ontologies is shown in Figure 1, the variables *ontobio* and *dwcterms*, on lines 6 and 7 respectively, store instances of ontology classes from a file and a web resource. BioDSL will load these ontologies into its RDF model.

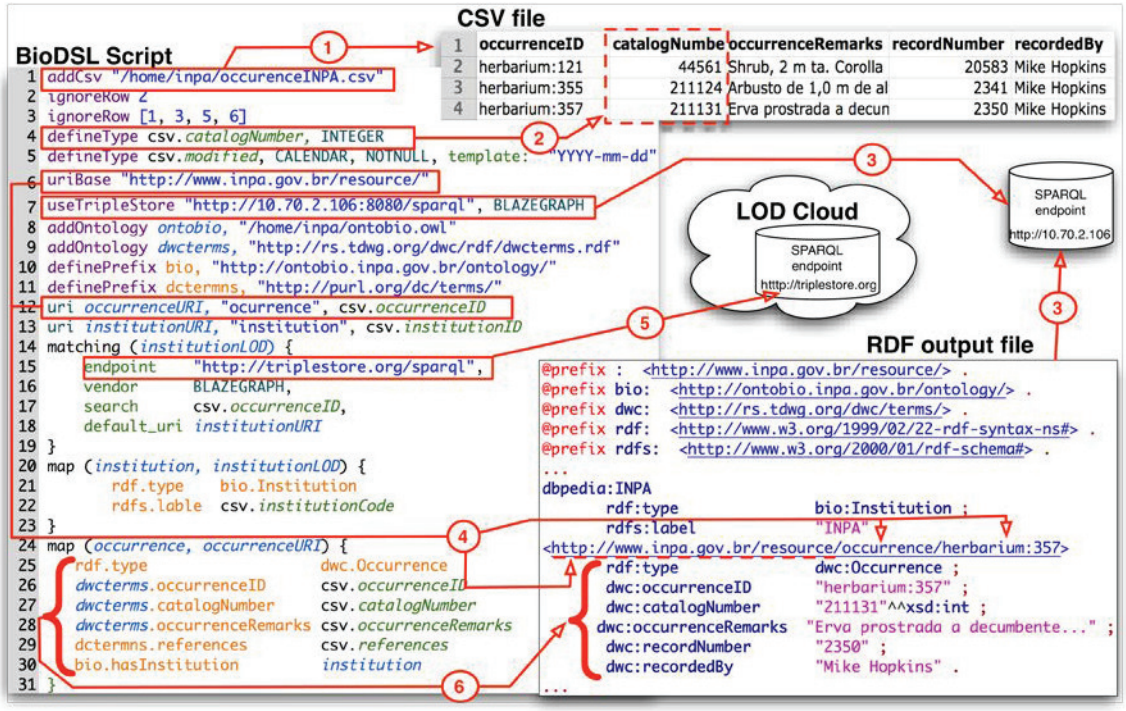

Figure 1. Configuring the CSV file and mapping to RDF in BioDSL

The *definePrefix* function, shown in Figure 1 lines 8 and 9, is used to simplify mapping ontology vocabularies to CSV entities. The objects in *bio* and *dwc* variables hold the ontology entities (classes, properties and individuals). These entities share the same prefix. In Figure 1, the *bio* and *dwc* variables are used latter in the mapping, matching and gazetteer functions.

3.3. Defining a Base URI

The *uriBase* function, Figure 1 line 10, allows users to set a base URI for the generated RDF. By default, individuals (resources) will be created, using this base URI. Unless an individual reuses a URI from the LOD supplied by the Matching and Gazetteer functions.

3.4. The *uri* function

The *uri* function (Figure 1, line 12) is used to build URIs of individuals to be used in the final RDF. To build a URI, the *uri* function gets the URI Base value and concatenates it with its parameters. The user can pass as parameters a string, column name, using the *csv* object, or a list of column names. In Figure 1, line 12, the function uses the string *occurrence* and the values from the *occurrenceID* column. Arrow 4.2 (Figure 1) indicates the final value in the RDF file.

3.5. *Matching* function

The *matching* function is the most relevant BioDSL function. It allows users to link their entities (from the CSV file) to entities from SPARQL endpoints. Then, using a semantic search engine, BioDSL can retrieve URIs from known entities in the endpoints to be reused to enrich data, during the RDF generation. Many entities from CSV files are already present in the LOD cloud, in sites such as Geonames, DBpedia, the SWI

Gazzetter[Cardoso et al. 2015]. Entity reuse can give the final RDF a high degree of interconnectivity to the LOD cloud. This may represent a way to the discovery of new knowledge.

The first parameter of *matching* function is its identifier. It is used by the *map* function to call a particular mapping when it needs it. The other matching functions parameter are all named. The *endpoint* parameter (Fig1. line 15) contains the SPARQL endpoint URL. The *vendor* parameter contains information about the SPARQL endpoint vendor. There are different strategies for differents vendors when BioDSL makes a semantic search. Currently, the BioDSL implementation supports two endpoint vendors, Blazegraph and Virtuoso. The *search* parameter contains a column name or set of column names (from the CSV file). The search is performed using the contents of the named column. If a set is used, the contents of each column are concatenated to form the search string. BioDSL uses the endpoints to searches string fields, such as labels, to find matching entities. If more than one entity is returned, the one with the best search score is used (different endpoints may use different search algorithms). If no entities are returned, the URI supplied by the *default* parameter is used. The optional parameter *type* is used to restrict the class (type) the entities, returned by the search string, should belong to. For instance, if the search string "Harpia harpyja" and type "http://dbpedia.org/ontology/species" are used, BioDSL will only return entities that belongs to this OWL class. This combination of best score and type restriction can be very effective in finding correct matchings, specially when using scientific terms, like species names. However, mistakes can happen and users should make tests to get acceptable error rates.

When the *matching* function retrieves an URI and reuses it in the final RDF as an entity name, BioDSL is saying that the entity, from the CSV file, and the entity, the URI represents, are the same. A link is created connecting the entity from the data set (being converted) to the LOD. Latter, that entity can be easily enriched using the facts already known about it in the LOD cloud.

3.6. Map function

The *map* function builds the final entities in RDF. It uses URIs, supplied by the user or generated by the *matching* and *uri*, and build RDF triples using the entities from the CSV file. Each triple add some information about the entity. For each data entity type, an user wants to map, a *map* function should be created.

The first parameter of the *map* function is its identifier. Using it, entities mapped by it can be referenced in other maps. In Figure 2, line 31, *institution* refers to the entity mapped by the *map*, in line 20. More specifically, to the institution referenced on the same line, in the CSV file, as the occurrence. By default, the *map* function identifier is also used to build the entity URI as the concatenation of the base URI (section 3.3), identifier and the line number in the CSV file. However, users can provide a *uri* or *matching* function identifier, as shown in Figure 2, labels "a" and "b". In this case, the URI returned by the functions will be used.

Once a URI is defined for the RDF entity, BioDSL maps its properties. In the function body, lines between the characters, users can define RDF property/values pairs. For property URIs and values (RDF object), they use prefix objects (created by a prefix

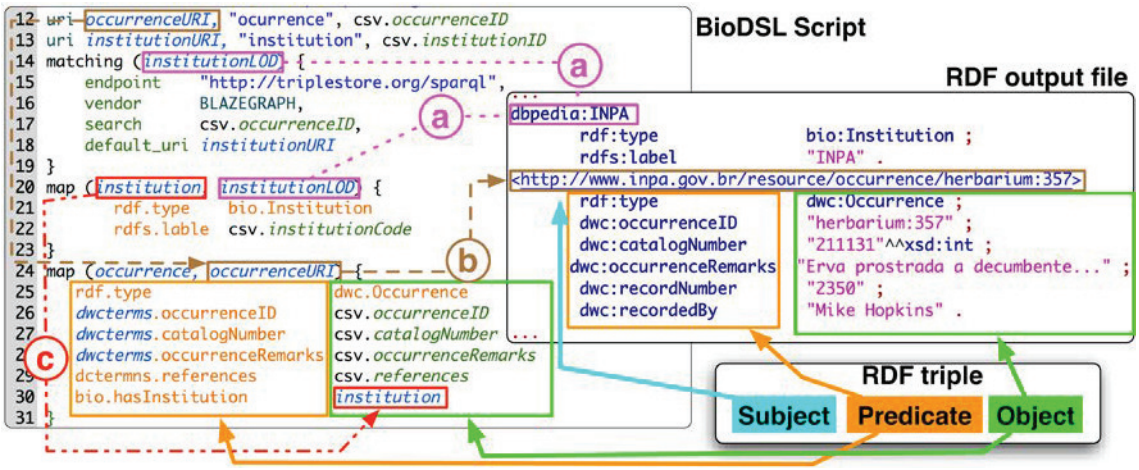

Figure 2. Details of how BioDSL generates RDF triples from maps, matches and uris: a) Using the matching function to find subject URIs from the LOD; b) Using the uri function to generate subject URL; c) Links between entities

function), shown in Figure 2 line 21, or a string representing a valid URI. Additionally, values can also use CSV column names (in this case, the column value is used, Figure 2 lines 22, 26 to 29) or an identifier of another map (Figure 2 label "c"). For each property/value pair, a triple is generated using the entity URI as subject, the property URI as predicate and the value as object.

4. Conclusion

In this paper, we presented a new approach for mapping open biodiversity data to RDF triples. It enables biodiversity users to easily take advantage of Semantic Web technologies, such as data enrichment with LOD information, data interlinking with known LOD entities, SPARQL endpoint, etc. This is all done in a less complicated way to users than other current mapping tools.

A small group of test users, involving experts in Semantic Web and biodiversity (from the National Institute for Amazonian Research - INPA), has been involved in the BioDSL development. For testing, it is being used open biodiversity data from brazilian institutions, users of the GBIF, SpeciesLink platform and SiBBR[16]. These data are in spreadsheet format and can be converted to CSV format, without loss of information. The current BioDSL implementation only supports data in CSV format.

A Web based IDE for BioDSL is under development. Users will be able to write BioDSL scripts with suggestions and feedback in a rich interactive environment. Users will be able to create, share, and publish these scripts and their data, as RDF triples (in a file or in a SPARQL endpoint) contributing to the LOD cloud growth.

5. Acknowledgment

We would like to thank ICMC-USP, LIS-INPA and CAPES (The National Council for the Improvement of Higher Education - Brazil), for partially supporting this work. Thanks are also due to our colleagues for ideas and for revising this text.

[16]http://www.sibbr.gov.br

References

Berners-Lee, T. (2006). Design issues: Linked data.

Cardoso, S. D., Amanqui, F. K., Serique, K. J. A., dos Santos, J. L. C., and Moreira, D. a. (2015). SWI: A Semantic Web Interactive Gazetteer to support Linked Open Data. *Future Generation Computer Systems*, pages –.

Dimou, A., Sande, M. V., Colpaert, P., Verborgh, R., Mannens, E., and Van De Walle, R. (2014). RML: A generic language for integrated RDF mappings of heterogeneous data. *CEUR Workshop Proceedings*, 1184.

Lebo, T. and Williams, G. T. (2010). Converting governmental datasets into linked data. In *Proceedings of the 6th International Conference on Semantic Systems - I-SEMANTICS '10*, page 1, New York, New York, USA. ACM Press.

Maali, F., Cyganiak, R., and Peristeras, V. (2011). Re-using Cool URIs: Entity Reconciliation Against LOD Hubs. *LDOW*.

Matsubara, W., Kusano, K., Bannai, H., and Shinohara, A. (2009). Language and Automata Theory and Applications. *Lata*, 5457:578–587.

Moura, A. M. D. C., Porto, F., Poltosi, M., Palazzi, D. C., Magalhães, P., and Vidal, V. (2012). Integrating Ecological Data Using Linked Data Principles. In *Joint V Seminar on Ontology Research in Brazil*, pages 156–167.

Scharffe, F., Atemezing, G., and Troncy, R. (2012). Enabling linkeddata publication with the datalift platform. *Proc. AAAI workshop on ...*.

Seyed, P., Chastain, K., Ashby, B., Liu, Y., Lebo, T., Patton, E., and McGuinness, D. (2013). Semanteco annotator. *CEUR Workshop Proceedings*, 1035:161–164.

Stadler, C., Unbehauen, J., Westphal, P., Sherif, M. A., and Lehmann, J. (2015). Simplified RDB2RDF Mapping. *Proceedings of the 8th Workshop on Linked Data on the Web (LDOW2015), Florence, Italy*.

Unbehauen, J., Hellmann, S., Auer, S., and Stadler, C. (2012). Knowledge extraction from structured sources. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7538:34–52.

Van Der Waal, S., Wecel Cel, K., Ermilov, I., Janev, V., Milosevic, U., and Wainwright, M. (2014). Lifting open data portals to the data web. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8661:175–195.

Walls, R. L., Deck, J., Guralnick, R., Baskauf, S., Beaman, R., Blum, S., Bowers, S., Buttigieg, P. L., Davies, N., Endresen, D., Gandolfo, M. A., Hanner, R., Janning, A., Krishtalka, L., Matsunaga, A., Midford, P., Morrison, N., Tuama, É. Ó., Schildhauer, M., Smith, B., Stucky, B. J., Thomer, A., Wieczorek, J., Whitacre, J., and Wooley, J. (2014). Semantics in Support of Biodiversity Knowledge Discovery: An Introduction to the Biological Collections Ontology and Related Ontologies. *PLoS ONE*, 9(3).