

# Improving discovery in Life Sciences Linked Open Data Cloud

Ali Hasnain<sup>1</sup>

Insight Center for Data Analytics, National University of Ireland, Galway  
`firstname.lastname@insight-centre.org`

**Abstract.** Multiple datasets that add high value to biomedical research have been exposed on the web as part of the Life Sciences Linked Open Data (LSLOD) Cloud. The ability to easily navigate through these datasets is crucial for personalized medicine and the improvement of drug discovery process. However, navigating these multiple datasets is not trivial as most of these are only available as isolated SPARQL endpoints with very little vocabulary reuse. The content that is indexed through these endpoints is scarce, making the indexed dataset opaque for users. We propose an approach to create an active Linked Life Sciences Data Compendium, a set of configurable rules which can be used to discover links between biological entities in the LSLOD cloud. We have catalogued and linked concepts and properties from 137 public SPARQL endpoints. Our Compendium is primarily used to dynamically assemble queries retrieving data from multiple SPARQL endpoints simultaneously.

## 1 Scene Setting

A considerable portion of the Linked Open Data cloud is comprised of datasets from Life Sciences Linked Open Data (LSLOD). The significant contributors includes the Bio2RDF project<sup>1</sup>, Linked Life Data<sup>2</sup> and the W3C HCLSIG Linking Open Drug Data (LODD) effort<sup>3</sup>. The deluge of biomedical data in the last few years, partially due to the advent of high-throughput gene sequencing technologies, has been a primary motivation for these efforts. There had been a critical requirement for a single interface, either programmatic or otherwise, to access the Life Sciences (LS) data. Although publishing datasets as RDF is a necessary step towards unified querying of biological datasets, it is not sufficient to retrieve meaningful information due to data being heterogeneously available at different endpoints [2]. Moreover in the LS domain, LD is extremely heterogeneous and dynamic [14,6]; also there is a recurrent need for *ad hoc* integration of novel experimental datasets due to the speed at which technologies for data capturing in this domain are evolving. As such, integrative solutions increasingly rely on federation of queries [4]. With the standardization of SPARQL 1.1, it is now possible to assemble federated queries using the “SERVICE” keyword, already supported by multiple tool-sets (SWobjects and Fuseki etc). To assemble queries encompassing multiple graphs distributed over different places, it is necessary

<sup>1</sup> <http://bio2rdf.org/> (l.a.: 2015-03-31 )

<sup>2</sup> <http://linkedlifedata.com/> (l.a.: 2015-05-16 )

<sup>3</sup> <http://www.w3.org/wiki/HCLSIG/LODD> (l.a.: 2014-07-16 )

that all datasets should be query-able using the same global schema [15]. This can be achieved either by ensuring that the multiple datasets make use of the same vocabularies and ontologies, an approach previously described as “*a priori integration*” or conversely, using “*a posteriori integration*”, which makes use of mapping rules that change the topology of remote graphs to match the global schema [5]. The methodology to facilitate the latter approach is the focus of our research. Moreover for LD to become a core technology in the LS domain, three challenges need to be addressed: *i*) dynamically discover datasets containing data regarding biological entities (e.g. Drugs, Molecules), *ii*) retrieve information about the same entities from multiple sources using different schemas, and *iii*) identify, for a given query, the highest quality data.

To address the aforementioned challenges, we introduce the notion of an active Compendium for LS data – a representation of entities and the links connecting these. Our methodology consisted of two steps: *i*) catalogue development, in which metadata is collected and analyzed, and *ii*) links creation, which ensures that concepts and properties are properly mapped to a set of Query Elements (*Qe*) [17]. For evaluation purposes, *Qe* are defined in the context of Drug Discovery and can be replaced by other *Qe(s)*. We assume that the proposed Compendium holds the potential to be used for a number of practical applications including assembling federated queries in a particular context. We already proposed the Link Creation mechanism, approaches and the linking statistics [7] as well as the cataloguing mechanism [9] and in this article we briefly report the methodology, initial results for Compendium development (cataloguing and linking), and an architecture for implementing Domain Specific Query Engine (under progress) as one of the practical applications that federates SPARQL query based on the set of mapping rules defined in the Compendium.

## 2 State of the Art

One approach to facilitate the “*A posteriori integration*” is through the use of available schema: semantic information systems have used ontologies to represent domain-specific knowledge and enable users to select ontology terms in query assembly [11]. BLOOMS, for example, finds schema-level links between LOD datasets using ontology alignment [10], but it relies mainly on Wikipedia. Ontology alignment typically relies on starting with a single ontology, which is not available for most SPARQL endpoints in the LOD cloud, hence could not be applied in our case. Furthermore, ontology alignment does not make use of domain rules (e.g. for two same sequences, qualifies for same gene) nor the use of URI pattern matching for alignment [7]. Approaches such as the VoID [1] and the SILK Framework [16] enable the identification of rules for link creation, but require extensive knowledge of the data prior to links creation. Query federation approaches have developed some techniques to meet the requirements of efficient query computation in the distributed environment. FedX [13], a project which extends the Sesame Framework [3] with a federation layer, enables efficient query processing on distributed LOD sources by relying on the assembly of a catalogue of SPARQL endpoints but does not use domain rules for links creation. Our approach for link creation towards Compendium development is a combination of

the several linking approaches as already explained by Hasnain et. al [7]: *i*) similarly to ontology alignment, we make use of label matching to discover concepts in LOD that should be mapped to a set of  $Qe$ , *ii*) we create “bags of words” for discovery of schema-level links similar to the approach taken by BLOOMS, and *iii*) as in SILK, we create domain rules that enable the discovery of links.

### 3 Proposed Approach/ Methodology

We proposed a Compendium for navigating the LSLOD cloud. Our methodology consists of two stages namely catalogue generation and link generation. Data was retrieved from 137 public SPARQL endpoints<sup>4</sup> and organized in an RDF document - the LSLOD Catalogue. The list of SPARQL endpoints was captured from publicly available Bio2RDF datasets and Datahub<sup>5</sup>.

#### 3.1 Methodology for Catalogue Development

For cataloguing, a preliminary analysis of multiple public SPARQL Endpoints was undertaken and a semi-automated method was devised to retrieve all classes (concepts) and associated properties (attributes) available by probing data instances. The workflow definition is as follows:

1. For every SPARQL endpoint  $S_i$ , find the distinct Classes  $C(S_i)$  :

$$C(S_i) = \text{Distinct} (\text{Project} (?class (\text{toList} (\text{BGP} (\text{triple} [] a ?class ))))) \quad (1)$$

2. Collect the Instances for each Class  $C_j(S_i)$  :

$$I_i : C_j(S_i) = \text{Slice} (\text{Project} (?I (\text{toList} (\text{BGP} (\text{triple} ?a a < C_j(S_i) > )))), \text{rand}()) \quad (2)$$

3. Retrieve the Predicate/Objects pairs for each  $I_i : C_j(S_i)$ :

$$I_i(P, O) = \text{Distinct} (\text{Project} (?p, ?o (\text{toList} (\text{BGP} (\text{triple} < I_i : C_j(S_i) > ?p ?o ))))) \quad (3)$$

4. Assign Class  $C_j(S_i)$  as domain of the Property  $P_k$  :

$$\text{Domain}(P_k) = C_j(S_i) \quad (4)$$

5. Retrieve Object type ( $O_T$ ) and assign as a range of the Property  $P_k$  :

$$\text{Range}(P_k) = O_T; O_T = \begin{cases} \text{rdf} : \text{Literal} & \text{if } (O_k \text{ is String}) \\ \text{dc} : \text{Image} & \text{if } (O_k \text{ is Image}) \\ \text{dc} : \text{InteractiveResource} & \text{if } (O_k \text{ is URL}) \\ \text{Project} (?R (\text{toList} (\text{BGP} \\ (\text{triple} < O_k > \text{rdf} : \text{type} ?R))) & \text{if } (O_k \text{ is IRI}) \end{cases} \quad (5)$$

RDFS, Dublin Core<sup>6</sup> and VoID<sup>7</sup> vocabularies were used for representing the data in the LSLOD catalogue, a slice of the catalogue for Pubchem SPARQL endpoint is presented (fig.1).

<sup>4</sup> <http://goo.gl/ZLbLzq>

<sup>5</sup> <http://datahub.io/> (l.a.: 2015-05-05)

<sup>6</sup> <http://dublincore.org/documents/dcmi-terms/> (l.a.: 2014-07-12)

<sup>7</sup> <http://vocab.deri.ie/void> (l.a.: 2014-07-12)

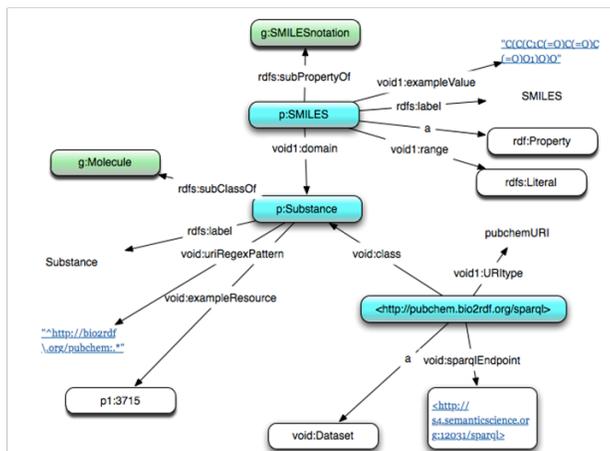


Fig. 1: An Extract from the LSLOD Catalogue for Pubchem dataset

### 3.2 Methodology for Link Generation

During this phase `subClassOf` and `subPropertyOf` links were created amongst different concepts and properties to facilitate “*a posteriori integration*”. The creation of links between identified entities (both chemical and biological) is not only useful for entity identification, but also for discovery of new associations such as protein/drug, drug/drug or protein/protein interactions that may not be obvious by analyzing datasets individually. Figure. 1 shows the `subClassOf` and `subPropertyOf` links with defined *Qes*. Links were created (discussed previously in [7]) using several approaches: *i*) Naïve Matching/ Syntactic Matching/ Label Matching, *ii*) Named Entity Matching, *iii*) Domain dependent/ unique identifier Matching, and *iv*) Regex Matching.

## 4 Applications/ Current Implementation

As of 31<sup>st</sup> May 2015, the Compendium consists of 280064 triples representing 1861 distinct classes and 3299 distinct properties catalogued from 137 endpoints.

### 4.1 DSQE: Domain-specific Query Engine

The general architecture of the DSQE (Fig 2) shows that given a SPARQL query, the first step is to parse the query and get individual triple patterns. Then Compendium is used for the triple pattern wise source selection (TPWSS) to identify relevant sources against individual triple patterns of the query. The Compendium enumerates the known endpoints relates each endpoint with one or more graphs and maps the local vocabulary to the vocabulary of the graph. The resulting query is executed on top of Apache Jena query engine.

An instance<sup>8</sup> of the DSQE is deployed in the context of drug discovery [8]. Using ‘Standard’ query builder, the user can select a topic of interest (e.g.

<sup>8</sup> <http://srvgal86.deri.ie:8000/graph/Granatum>

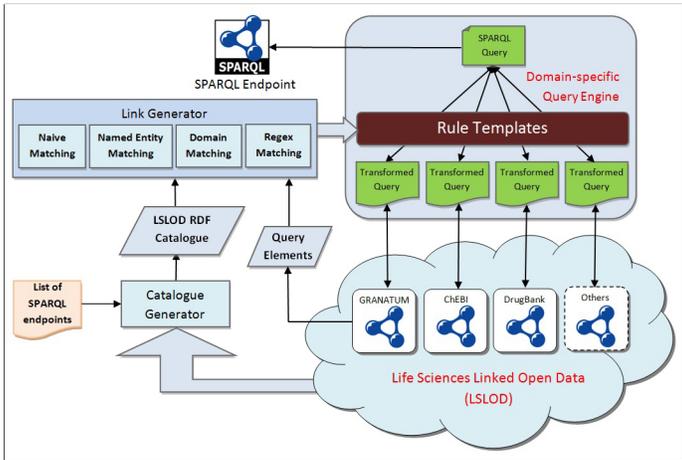


Fig. 2: Compendium and Query Engine Architecture

Molecule) along with the list of associated properties. We plot the catalogued subclasses of few  $Q_e$  and the total number of distinct instances retrieved per  $Q_e$  while querying using DSQE (Fig 3).

## 5 Evaluation

So far we evaluated the performance of our catalogue generation methodology and recorded the times taken to probe instances through endpoint analysis of 12 endpoints whose underlying data sources were considered relevant for drug discovery - Medicare, Dailymed, Disasome, DrugBank, LinkedCT, Sider, National Drug Code Directory (NDC), SABIO-RK, Saccharomyces Genome Database (SGD), KEGG, ChEBI and Affymetrix probesets. The cataloguing experiments were carried out on a standard machine with 1.60Ghz processor, 8GB RAM using a 10Mbps internet connection. We recorded the total available concepts and properties at each SPARQL endpoint as well as those actually catalogued in our Compendium [9]. Total number of triples exposed at each of these SPARQL endpoints and the time taken for cataloguing was also recorded. We selected those SPARQL endpoints which have a better latency for this evaluation, as the availability and the uptime of the SPARQL endpoint is an important factor for cataloguing. Best fit regression models were then calculated. As shown in Fig. 4, our methodology took less than 1000000 milliseconds (<16 minutes) to catalogue seven of the SPARQL endpoints, and a gradual rise with the increase in the number of available concepts and properties. We obtained two power regression models ( $T = 29206 * C_n^{1.113}$  and  $T = 7930 * P_n^{1.027}$ ) to help extrapolate time taken to catalogue any SPARQL endpoint with a fixed set of available concepts ( $C_n$ ) and properties ( $P_n$ ), with  $R^2$  values of 0.641 and 0.547 respectively. Using these models and knowing the total number of available concepts/properties, a developer could determine the approximate time (ms) as a vector combination. KEGG and SGD endpoints took an abnormally large amount of time for cataloguing than the trendline. The reason for this may include endpoint timeouts or network

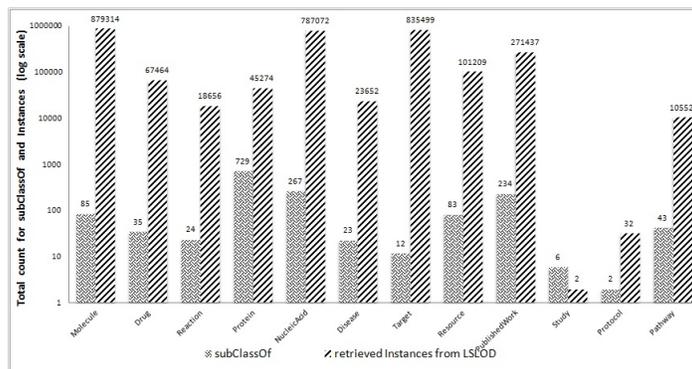


Fig. 3: Number of retrieved Instances and Subclasses linked to any  $Q_e$

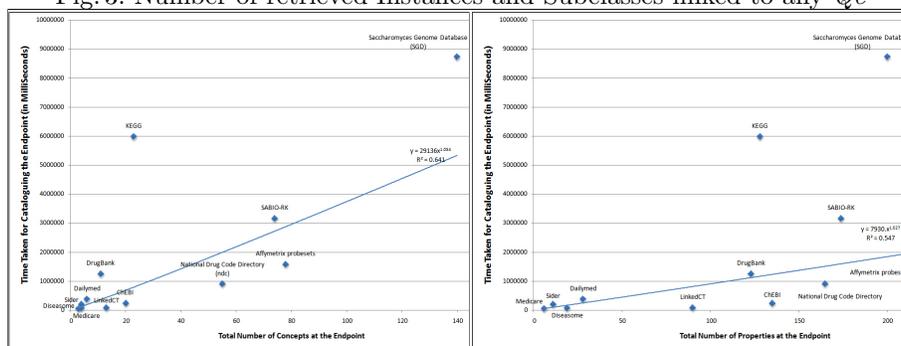


Fig. 4: Time taken to catalogue 12 SPARQL endpoints

delays. We also evaluated the performance of our Link Generation methodology by comparing it against the popular linking approaches. Using WordNet thesauri we attempted to automate the creation of bags of related words using 6 algorithms [7]: Jing Conrath, Lin, Path, Resnik, Vector and WuPalmer with unsatisfactory results (Figure 5(c)). Our linking approaches resulted in better linking rate as shown in Figure 5(a,b)

## 6 Discussion

There is great potential in using semantic web and LD technologies for accessing and querying Life sciences data for finding Meaningful Biological Correlations. However, in most cases, it is not possible to predict *a priori* where the relevant data is available and its representation. Our current research provides the concept and methodology for devising an active Linked Life Sciences Data Compendium that relies on systematically issuing queries on various life sciences SPARQL endpoints and collecting its results in an approach that would otherwise have to be encoded manually by domain experts. Current experiments and evaluation uses a set of  $Q_e$ , which were defined in a context of drug discovery. The number of classes per endpoint varied from a single class to a few thousands. Our initial exploration of the LSLOD revealed that only 15% of classes are reused. However, this was not the case for properties, of which 48.5% are reused. Most

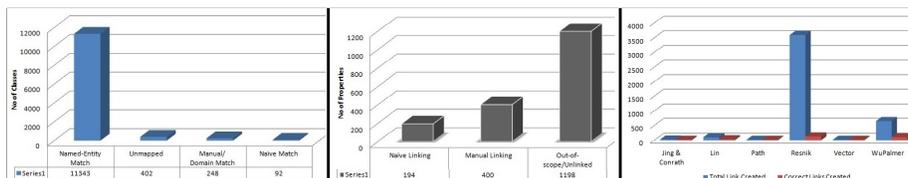


Fig. 5: (a) Number of Classes Linked, (b) Number of Properties Linked, (c) Number of Classes linked through available similarity linking approaches

of the properties found were domain independent (e.g. `type`, `seeAlso`); however, these are not relevant for the Compendium as they cannot increase the richness of information content. Although a very low percentage of linking becomes possible through naïve matching or manual/domain matching, the quality of links created are highly trusted [7]. It is also worth noticing that 23% of identified classes, and 56.2% of the properties remained unlinked, either because they are out of scope or cannot match any  $Qe$ . This means that the quality as well as the quantity of links created is highly dependent on the set of  $Qe$  used.

## 7 Open Issues and Future Directions

Multiple challenges faced which can hinder the applicability of our approach:

- Some endpoints return timeout errors when a simple query (`SELECT DISTINCT ?Concept WHERE { [ ] a ?Concept }`) is issued.
- Some endpoints have high downtime and cannot be generally relied.
- Many endpoints provide non-deferenceable URI and some deferenceable URI do not provide a “type” for the instance.

In future an extension under consideration to available Compendium is to enrich it with statistical and provenance information with appropriate changes to DSQE and evaluate the overall performance. This includes information including `void:triples`, `void:entities`, `void:classes`, `void:properties`, `void:distinctSubjects` and `void:distinctObjects` in case of statistical cataloguing where as `dcterms:title`, `dcterms:description`, `dcterms:date`, `dcterms:publisher`, `dcterms:contributor`, `dcterms:source`, `dcterms:creator`, `dcterms:created`, `dcterms:issued` and `dcterms:modified` in case of provenance. Currently we are extending DSQE to convert any SPARQL 1.0 query into corresponding SPARQL 1.1 query by using TPWSS information and the SPARQL “SERVICE” clause. Implementing so DSQE will be able to answer any federated SPARQL Query considering the desired endpoint being catalogued in Compendium. The performance of this extended DSQE is aimed to compare with state of the art query Engine FedX [13] using extensive evaluation criteria including *source selection in terms of number of ASK*, *total triple pattern-wise sources selected*, *source selection time* and *total number of results retrieved per query*. For this evaluation we aim to select some queries from available query federation benchmark e.g FedBench [12] and also plan to define some complex biological queries applicable on 10 real time publicly available datasets. Issues related to Identity Resolution are also considered as future work.

## Acknowledgement

This research has been supported in part by Science Foundation Ireland under Grant Number SFI/12/RC/2289 and SFI/08/CE/I1380 (Lion 2). The author would also like to acknowledge Stefan Decker being PhD supervisor.

## References

1. Alexander, K., Hausenblas, M.: Describing linked datasets-on the design and usage of void, the'vocabulary of interlinked datasets. In: In Linked Data on the Web Workshop (LDOW 09), in conjunction with WWW09. Citeseer (2009)
2. Bechhofer, S., Buchan, I., De Roure, D., Missier, P., et al.: Why linked data is not enough for scientists. *Future Generation Computer Systems* 29(2), 599–611 (2013)
3. Broekstra, J., Kampman, A., Van Harmelen, F.: Sesame: A generic architecture for storing and querying RDF and RDF schema. In: *The Semantic Web—ISWC 2002*, pp. 54–68. Springer (2002)
4. Cheung, K.H., Frost, H.R., Marshall, M.S., et al.: A journey to semantic web query federation in the life sciences. *BMC bioinformatics* 10(Suppl 10), S10 (2009)
5. Deus, H.F., Prud'hommeaux, E., Miller, M., Zhao, J., Malone, J., Adamusiak, T., et al.: Translating standards into practice—one semantic web API for gene expression. *Journal of biomedical informatics* 45(4), 782–794 (2012)
6. Goble, C., Stevens, R., Hull, D., et al.: Data curation+ process curation= data integration+ science. *Briefings in bioinformatics* 9(6), 506–517 (2008)
7. Hasnain, A., Fox, R., Decker, S., Deus, H.F.: Cataloguing and linking life sciences LOD Cloud. In: *1st International Workshop on Ontology Engineering in a Data-driven World collocated with EKAW12* (2012)
8. Hasnain, A., Kamdar, M.R., Hasapis, P., Zeginis, D., Warren Jr, C.N., et al.: Linked Biomedical Dataspace: Lessons Learned integrating Data for Drug Discovery. In: *International Semantic Web Conference (In-Use Track)*, October 2014 (2014)
9. Hasnain, A., e Zainab, S.S., Kamdar, M.R., Mehmood, Q., Warren Jr, C.N., Fatimah, Q.A., Deus, H.F., Mehdi, M., Decker, S.: A roadmap for navigating the life sciences linked open data cloud. In: *Semantic Technology*, pp. 97–112. Springer (2014)
10. Jain, P., Hitzler, P., Sheth, A.P., Verma, K., Yeh, P.Z.: Ontology alignment for linked open data. In: *The Semantic Web—ISWC 2010*, pp. 402–417. Springer (2010)
11. Petrovic, M., Burcea, I., Jacobsen, H.A.: S-ToPSS: semantic toronto publish/subscribe system. In: *Proceedings of the 29th international conference on Very large data bases—Volume 29*. pp. 1101–1104. *VLDB Endowment* (2003)
12. Schmidt, M., Görlitz, O., Haase, P., Ladwig, G., Schwarte, A., Tran, T.: Fedbench: A benchmark suite for federated semantic data query processing. In: *The Semantic Web—ISWC 2011*, pp. 585–600. Springer (2011)
13. Schwarte, A., Haase, P., Hose, K., Schenkel, R., Schmidt, M.: Fedx: a federation layer for distributed query processing on linked open data. In: *The Semantic Web: Research and Applications*, pp. 481–486. Springer (2011)
14. Stein, L.D.: Integrating biological databases. *Nature Reviews Genetics* 4(5), 337–345 (2003)
15. Studer, R., Grimm, S., Abecker, A.: *Semantic web services: concepts, technologies, and applications*. Springer (2007)
16. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: *Discovering and maintaining links on the web of data*. Springer (2009)
17. Zeginis, D., et al.: A collaborative methodology for developing a semantic model for interlinking Cancer Chemoprevention linked-data sources. *Semantic Web* (2013)