Making Linked Data SPARQL with the InterMine Biological Data Warehouse

Maxime Déraspe[1,2], Gail Binkley[5], Daniela Butano[3,4], Matthew Chadwick[3,4], J. Michael Cherry[5], Justin Clark-Casey[3,4], Sergio Contrino[3,4], Jacques Corbeil[1], Josh Heimbach[3,4], Kalpana Karra[5], Rachel Lyne[3,4], Julie Sullivan[3,4], Yo Yehudi[3,4], Gos Micklem[3,4], and Michel Dumontier[2]

[1] Department of Molecular Medicine, Université Laval, Québec, Canada
[2] Stanford Center for Biomedical Informatics Research, Stanford University, Stanford, United States
[3] Department of Genetics, University of Cambridge, Cambridge, United Kingdom
[4] Cambridge Systems Biology Centre, University of Cambridge, Cambridge, United Kingdom
[5] Department of Genetics, Stanford University, Stanford, United States

Abstract. InterMine is a system for integrating, analysing, and republishing biological data from multiple sources. It provides access to these data via a web user interface and programmatic web services. However, the precise invocation of services and subsequent exploration of returned data require substantial expertise on the structure of the underlying database. Here, we describe an approach that uses Semantic Web technologies to make InterMine data more broadly accessible and reusable, in accordance with the FAIR principles. We describe a pipeline to extract, transform, and load a Linked Data representation of the InterMine store. We use Docker to bring together SPARQL-aware applications to search, browse, explore, and query the InterMine-based data. Our work therefore extends interoperability of the InterMine platform, and supports new query functionality across InterMine installations and the network of open Linked Data.

Keywords: linked data, SPARQL, RDF, biological data warehouse, integrative bioinformatics

1 Introduction

InterMine is a Java-based open-source data warehouse created specifically for the integration and analysis of biological information [1]. It can load data from a wide range of heterogeneous data sources into a data model that is mutable and extensible, and expose this loaded data in a manner that is easy to explore and mine. Many MODs (model organism databases) use the InterMine platform to make their data available to users [2], such as the MODs for fly [3], mouse [4], nematode [5], rat [6], budding yeast [7] and zebrafish [8]. It is also in use in many other projects such as modENCODE [9] and for drug discovery [10].

In order to implement its flexible data model, InterMine stores data using a custom Object Relational Mapping (ORM) in a PostgreSQL database. Data objects are presented to the user via a web interface and via REST-ful web services and clients that implement a bespoke API[11]. These access mechanisms are comparable with other primary and secondary biological databases [12].

Integration of an arbitrary number of data sources into a single system is one of InterMine's primary features. However, users may still want to perform further integration with sources that remain outside the data warehouse. For instance, they may have additional unpublished or private datasets; a data source may be integrated with InterMine but not to the level of detail that they require; or they may require extensive ad-hoc cross-domain data integration in the course of their research that is difficult to anticipate.

In this case, the integration benefits of InterMine are reduced. Users have to fall back to performing further manual integration, which is difficult and time consuming due to differences between file formats and data-access services provided by InterMine and other data sources [12]. Manual integration also incurs maintenance costs over time as data formats and access services evolve [13].

Over recent years, various data providers, notably the European Bioinformatics Institute (EBI) [14] and PubChem [15], have started to provide their data as RDF Linked Data in addition to their existing data-access facilities. Providing information in a common structured form allows a user to download datasets from one or more sources and perform queries across them using standard SPARQL query mechanisms. When organizations such as the EMBL-EBI provide a public SPARQL endpoint [14], these queries can also be performed directly over the Internet, potentially across many different data providers at once.

Providing Linked Data also advances FAIR (Findable, Accessible, Interoperable, and Re-usable) principles [16], a vision that lies at the heart of the InterMine project. Therefore, we are extremely interested in how we can implement a process to make it easy for an InterMine operator to provide RDF Linked Data and a public SPARQL endpoint as an extension of the InterMine system.

In this paper, we describe a very important component of this process, namely a mechanism created recently by the Dumontier Lab at Stanford University to generate Linked Data from InterMine-loaded data. We also describe the same lab's Model Organism Linked Database (MOLD), a Linked Open Data cloud generated from the RDF output of six MOD InterMine installations [17].

Following on from this, we discuss future work by which we could adapt this RDFization mechanism to allow any InterMine operator to easily generate Linked Data and make it downloadable and queryable. We will talk about the process and challenges involved, both in terms of data and in terms of technology.

2   Converting InterMine data: RDFization

The InterMine-RDFizer [18] is an open-source software tool that allows a user to generate RDF Linked Data from data loaded into InterMine. The tool works

by extracting data from InterMine using its standard web services. This is in contrast to projects such as D2RQ [19] that directly map relational tables to RDF graphs. In experimental work we have found it difficult to adapt such projects to InterMine's custom ORM database structure, where data objects are split over multiple tables generated from a mutable data model. By contrast, the InterMine-RDFizer receives logical unified views of the data objects, which are much easier to convert to the RDF data model.

Figure 1 shows the implementation view of the InterMine-RDFization process, where data is downloaded into Tab Separated Value (TSV) files and then converted into RDF triples.
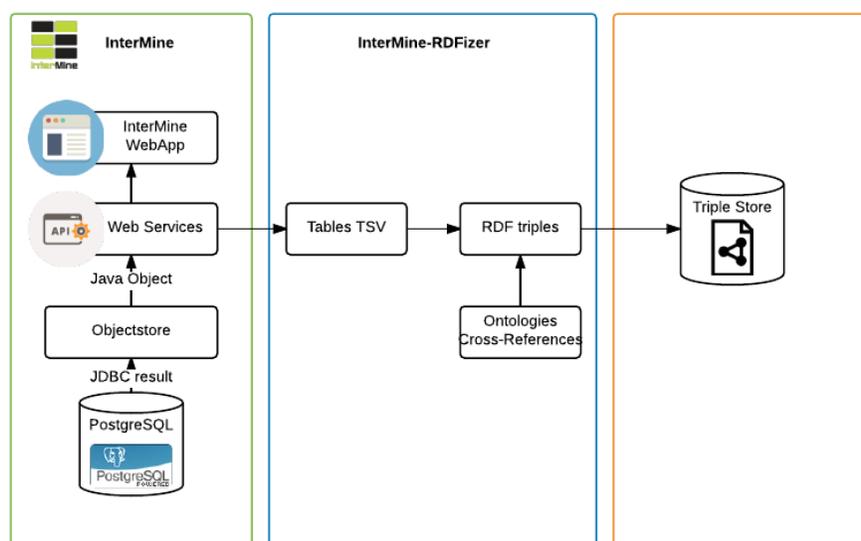

Fig. 1. InterMine-RDFization Process

InterMine stores data as representations of biological objects (Genes, Organisms, Proteins, etc.) in a class-based model. The InterMine-RDFizer maps each biological object to an RDF resource. The resource type is based on the class name (e.g. Gene, Organism) and the resource URI is built using the unique sequential ID assigned by InterMine's ORM system to each object when it is loaded. The listing 1.1, for example, represents the triples generated for the gene with ID 1007664:

```
<http://mo-ld.org/flymine:1007664> rdf:type http://mo-ld.org/resource/
    flymine_SequenceFeature>
<http://mo-ld.org/flymine:1007664> rdf:type http://mo-ld.org/resource/flymine_Gene
    >
```

Listing 1.1. Resource types created in the RDFization process

The InterMine-RDFizer generates predicates using a generic approach from the properties of each InterMine data class. Figure 2, for instance, partially shows the resources generated for the gene with symbol "zen" in the organism *Drosophila melanogaster*.
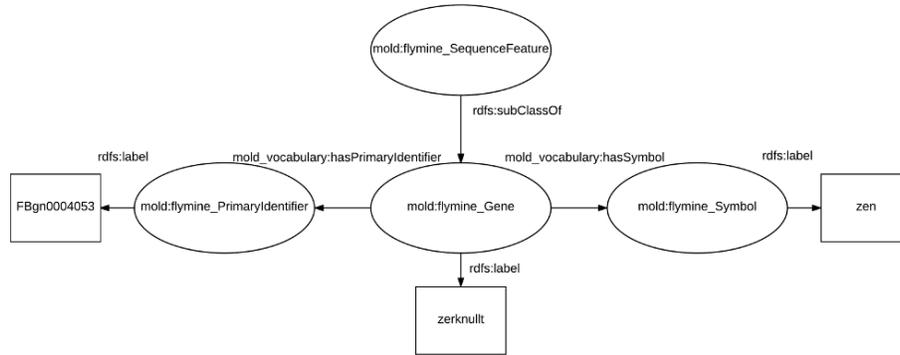

Fig. 2. Resources generated by the RDFization process

Importing the created triples into a triplestore allows a user to query the generated data using SPARQL, the standard query language for RDF. Query 1.2 shows how one can fetch genes from the organism *Drosophila melanogaster* annotated with a specified GO term, using the triples generated by the InterMine-RDFizer from the FlyMine MOD InterMine installation.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX mold: <http://mo-ld.org/resource/>
PREFIX mold_voc: <http://mo-ld.org/mine_vocabulary:>
SELECT DISTINCT ?primaryIdentifier ?symbol ?termIdentifier ?termName
WHERE {
  ?gene a mold:flymine_Gene;
        mold_voc:hasOrganism/rdfs:label ?organism .
  FILTER (?organism="Drosophila_melanogaster") .
  ?gene mold_voc:hasPrimaryIdentifier/rdf:value ?primaryIdentifier;
        mold_voc:hasSymbol/rdf:value ?symbol;
        mold_voc:hasGOAnnotation/mold_voc:hasOntologyTerm ?term .
  ?term rdfs:label ?termName .
  FILTER (?termName="nucleoplasm") .
  ?term mold_voc:hasIdentifier/rdf:value ?termIdentifier
}
```

Query 1.2. SPARQL query for genes annotated with a specified GO term

3 Creating Linked Data

As part of its data integration process, InterMine merges data from multiple sources into common data objects. For instance, a protein object may contain

data from UniProt merged with records from other protein data sources like IntAct or InterPro. For any merged source, InterMine stores cross-references to other databases (e.g. PubMed cross-references in InterPro data) in a cross-references table.

The InterMine-RDFizer uses these stored identifiers to generate Linked Data. The script has to be provided with a file containing the mapping between the data source name, as stored in InterMine (e.g. UniProt), and the URI of the external RDF repository (e.g. `http://purl.uniprot.org/uniprot/`). For example, the protein "Breast cancer type 1 susceptibility protein", in the organism "Homo sapiens", could be linked to the resource `<http://purl.uniprot.org/uniprot/BRCA1_HUMAN>`. In addition to cross-references, the RDFizer can also link entries in InterMine's ontology tables to external ontology term (class) URLs, using the same configuration file. For instance, the Gene Ontology term with identifier GO:0005654 could be linked to the resource `http://amigo.geneontology.org/amigo/term/GO:0005654`.

4 MOLD project

The InterMine-RDFizer was developed as part of the MOLD (Model Organism Linked Data) project. This is a Semantic Web platform, recently developed by the Dumontier Lab at Stanford University, for publishing model organism data under FAIR[16] principles. It currently includes 6 MODs: FlyMine, HumanMine, MouseMine, YeastMine, RatMine, ZebrafishMine.

MOLD uses the RDFizer to generate RDF from the InterMine installations of these MODs. The RDFizer also links this generated data to Bio2RDF [20][21][22], one of the largest networks of Linked Data for the life sciences. The data is also linked to external ontologies as such as GO [23] and the Sequence Ontology [24].

The MOLD platform provides a web interface to query, browse and exploring its contained RDF data. This includes a SPARQL editor with a result viewer supporting several result set formats, a search widget providing a full text search, and the RelFinder tool [25], to interactively explore relations between two RDF resources, for which some examples have been already provided.

In addition, the MOLD platform provides a REST-based web services API that currently supports the following commands: describe, links, search and sparql. The example below 1.3 shows how to retrieve the triples that describe a resource given the resource URI `http://mo-ld.org/flymine:1007664`

```
curl -X GET --header 'Accept: text/html' 'http://api.mo-ld.org:80/v1/describe?uri=
    http%3A%2F%2Fmo-ld.org%2Fflymine%3A1007664'
```

Listing 1.3. HTTP request to describe endpoint

The user can access the same API via the web interface, editing the input parameters and browsing the returned results.

The Docker container system is used to deploy the MOLD project. Docker packages a software application together with its dependencies in a single image,

eliminating the need to separately install other software libraries and frameworks. The MOLD project provides three images: one for the triple store, one for the MOLD web application and one for the REST API.

5   Future work

As we have seen, the InterMine-RDFizer facility can generate RDF from an InterMine installation and the MOLD project used this to create a Linked Open Data network from 6 MOD InterMine instances. Our interest now is to extend this work so that we can ship RDF generation and SPARQL query facilities as a native component of the InterMine system. We want to do this in such a way that any operator can activate and maintain these facilities without major operational overhead, no matter what type of data their mine integrates. This will make more RDF and SPARQL endpoints available for InterMine-integrated datasets, and give a reasonable expectation that generated RDF data will remain in sync with InterMine-loaded data.

To achieve these goals, we need to tackle a number of challenges. On the data side, we need to make sure that any InterMine resource in the generated RDF has an IRI that is unique and stable over time, one of the core Linked Data requirements [26]. This is not straightforward because, unlike primary biological databases, InterMine does not have prior knowledge of the structure of loaded biological data, as its core data model is mutable and extensible.

Currently, the InterMine-RDFizer script generates resource IRIs that use the sequential IDs that InterMine generates as part of its ORM system (e.g. `http://mo-ld.org/flymine:1007664`). These will not be stable over time since these IDs will change when the data in the warehouse is updated. Instead, for each externally referenceable data class we may need to identify which properties form a unique and temporally-stable key. One possibility is to concatenate the data class name (e.g. "Protein") with a primary ID property that comes from one of the loaded external data sources (e.g. P38398 from Uniprot).

Regarding Linked Data, we also want to ensure that the RDF generated from any particular InterMine object links back to the data sources that were integrated into that object. As we described in an earlier section, the InterMine-RDFizer uses InterMine's cross-reference and ontology term data to generate links. However, these capture cross-references provided by the source rather than the source itself (e.g. we are not generating triples that link an InterPro IRI to an InterMine ProteinDomain resource). Capturing this data for RDF generation may require some additional data source recording by InterMine itself.

Another data-related challenge concerns ontologies. The data sources that are loaded into InterMine often use ontology terms as property values, such as Gene Ontology terms to identify the functions of a gene. However, except in an automated fashion for sequence properties, InterMine does not attach ontological terms to the properties themselves. For instance, properties such as "abstractText" and "title" in the Publication class are not attached to terms in the Dublin Core ontology.

The InterMine-RDFizer handles this by automatically generating RDF predicates from InterMine property names. For example, it generates the IRI `http://mo-ld.org/mine_vocabulary:hasAuthor` for the "authors" property of the Publication class. But we would also want to make it possible to use predicates from existing ontologies, such as those from the Dublin Core ontology above. We would need to either extend InterMine itself to associate ontology terms with data model properties, or provide a further configuration mechanism in the InterMine-RDFizer that can do this at RDF generation time.

On the technological side, a major issue concerns how InterMine data converted to RDF will be stored and made available to users. The Docker images created by MOLD that provide a triplestore, web application and REST API will serve as a very useful base. We will need to assess the performance, ease of use and maintainability of the systems used, in the context of making this a very generic facility for any InterMine installation.

We will also want to integrate RDF downloads with the InterMine web interface proper. InterMine has existing facilities for exporting data in different formats (CSV, JSON, etc.), so adding a further option to link to the URI that serves RDF for a particular biological object would be very desirable. This architectural layout is shown in figure 3.
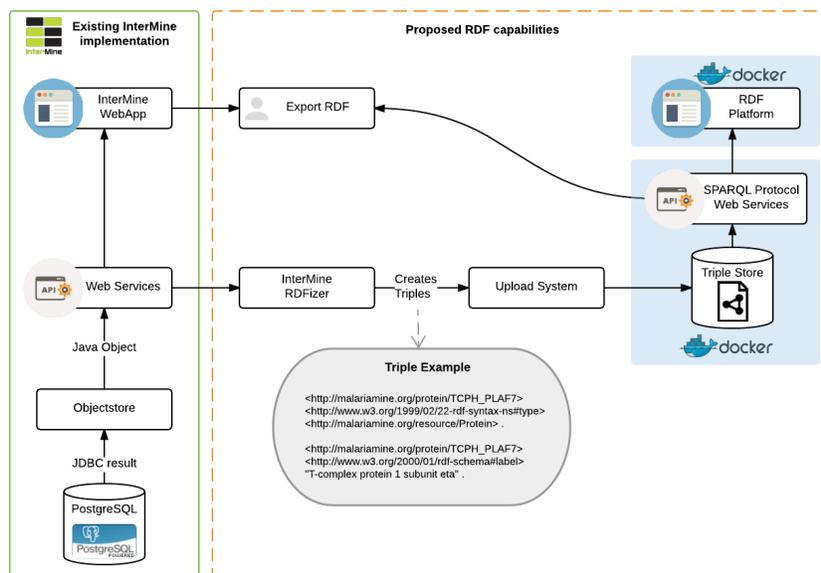

Fig. 3. InterMine RDF Provision Process

6 Acknowledgements

References

1. Smith RN, Aleksic J, Butano D, Carr A, Contrino S, Hu F, Lyne M, Lyne R, Kalderimis A, Rutherford K, Stepan R, Sullivan J, Wakeling M, Watkins X, Micklem G. InterMine: a flexible data warehouse system for the integration and analysis of heterogeneous biological data. Bioinformatics. 28(23):3163-5 (2012)
2. Lyne R, Sullivan J, Butano D, Contrino S, Heimbach J, Hu F, Kalderimis A, Lyne M, Smith RN, Štěpán R, Balakrishnan R, Binkley G, Harris T, Karra K, Moxon SA, Motenko H, Neuhauser S, Ruzicka L, Cherry M, Richardson J, Stein L, Westerfield M, Worthey E, Micklem G. Cross-organism analysis using InterMine. Genesis. 53(8):547-60 (2015)
3. Lyne R, Smith R, Rutherford K, Wakeling M, Varley A, Guillier F, Janssens H, Ji W, Mclaren P, North P, Rana D, Riley T, Sullivan J, Watkins X, Woodbridge M, Lilley K, Russell S, Ashburner M, Mizuguchi K, Micklem G. FlyMine: an integrated database for Drosophila and Anopheles genomics. Genome Biol. 8(7):R129 (2007)
4. Motenko H, Neuhauser SB, O'Keefe M, Richardson JE. MouseMine: a new data warehouse for MGI. Mamm Genome. 26(7-8):325-30 (2015)
5. Howe KL, Bolt BJ, Cain S, Chan J, Chen WJ, Davis P, Done J, Down T, Gao S, Grove C, Harris TW, Kishore R, Lee R, Lomax J, Li Y, Muller HM, Nakamura C, Nuin P, Paulini M, Raciti D, Schindelman G, Stanley E, Tuli MA, Van Auken K, Wang D, Wang X, Williams G, Wright A, Yook K, Berriman M, Kersey P, Schedl T, Stein L, Sternberg PW. WormBase 2016: expanding to enable helminth genomic research. Nucleic Acids Res. 44(D1):D774-80 (2016)
6. http://ratmine.org/
7. Balakrishnan R, Park J, Karra K, Hitz BC, Binkley G, Hong EL, et al. YeastMine– an integrated data warehouse for Saccharomyces cerevisiae data as a multipurpose tool-kit. Database . 2012: bar062 (2012)
8. http://zebrafishmine.org/
9. Contrino S, Smith RN, Butano D, Carr A, Hu F, Lyne R, et al. modMine: flexible access to modENCODE data. Nucleic Acids Res. 40: D1082–8 (2012)
10. Chen Y-A, Yi-An C, Tripathi LP, Kenji M. An integrative data analysis platform for gene set analysis and knowledge discovery in a data warehouse framework. Database. 2016: baw009 (2016)
11. Kalderimis A, Lyne R, Butano D, Contrino S, Lyne M, Heimbach J, Hu F, Smith R, Stěpán R, Sullivan J, Micklem G. InterMine: extensive web services for modern biology. Nucleic Acids Res. 42(Web Server issue):W468-72 (2014)
12. Stein LD. Integrating biological databases. Nat Rev Genet. 4: 337–345 (2003)
13. Goble C, Carole G, Robert S. State of the nation in data integration for bioinformatics. J Biomed Inform. 41: 687–693 (2008)

14. Jupp S, Malone J, Bolleman J, Brandizi M, Davies M, Garcia L, et al. The EBI RDF platform: linked open data for the life sciences. Bioinformatics. 30: 1338–1339 (2014)

15. Fu G, Batchelor C, Dumontier M, Hastings J, Willighagen E, Bolton E. PubChem-RDF: towards the semantic annotation of PubChem compound and substance databases. J Cheminform. 7: 34 (2015)

16. Wilkinson MD, Dumontier M, Aalbersberg IJJ, Appleton G, Axton M, Baak A, et al. The FAIR Guiding Principles for scientific data management and stewardship. Sci Data. 3: 160018 (2016)

17. http://mo-ld.org/

18. https://github.com/mo-ld/intermine-rdfizer

19. http://d2rq.org/

20. Belleau F, Nolin MA , Tourigny N, Rigault P, andMorissette J. Bio2RDF: towards a mashup to build bioinformatics knowledge systems.Journal of biomedical informatics, vol. 41, pp. 706–16 (2008)

21. Nolin MA, Ansell P, Belleau F, Idehen K, Rigault P, Tourigny N, Roe P, Hogan JM, and Dumontier M. Bio2RDF network of linked data. Semantic Web Challenge; International Semantic Web Conference (ISWC 2008). Citeseer (2008)

22. Callahan A, Cruz-Toledo J, Ansell P, Dumontier M, Bio2RDF Release 2: Improved Coverage, Interoperability and Provenance of Life Science Linked Data, in The Semantic Web: Semantics and Big Data (Cimiano P, Corcho O, Presutti V, Hollink L, and Rudolph S, eds.), vol. 7882 of Lecture Notes in Computer Science, pp. 200–212, Springer Berlin Heidelberg (2013)

23. The Gene Ontology Consortium. Gene Ontology Consortium: going forward. Nucl Acids Res 43 Database issue D1049–D1056 (2015)

24. Eilbeck K., Lewis S., Mungall C.J., Yandell M., Stein L., Durbin R., Ashburner M. The Sequence Ontology: A tool for the unification of genome annotations. Genome Biology 6:R44 (2005)

25. http://www.visualdataweb.org/relfinder.php

26. Hausenblas M. 5 * Open Data [Internet]. [cited 12 Sep 2016]. Available: http://5stardata.info/en/ (2016)