

Collaborative development of cross-database Bio2RDF queries

Peter Ansell¹

¹Queensland University of Technology, Brisbane, Australia, p.ansell@qut.edu.au

ABSTRACT

The Bio2RDF server can be used to perform cross-database queries relating to both biological databases and other unrelated databases. The server is configured using an RDF model that enables users to extend the global configuration used by the <http://bio2rdf.org/> mirror servers, or create an entirely new configuration. In order to allow users to collaborate easily, a web application has been created that enables users to create and edit the queries, providers, and namespaces that form the basis of the model. Users authenticate to the web application using OpenID, and their OpenID URL's are used to attribute different parts to their authors. Users can use their additions by configuring their local installations of the Bio2RDF server using URL's provided by the configuration web application, or by copying the relevant RDF code to their server.

INTRODUCTION

Modern biological datasets are characterised by their size, the intricacies of the inter-relationships among their elements, and by the variations in annotation and labelling for similar and even for identical data items [1]. In order to solve this issue the Bio2RDF project created RDF versions of each of the major biological databases, together with simple mappings where possible between different databases to identify identical items [2]. These RDF versions could be loaded into a single RDF database, but in order to make public endpoints available, and keep them maintainable, it is easier to load them into separate RDF databases and query databases as needed.

The majority of semantic web efforts related to biology are focused on being able to execute SPARQL queries on single RDF databases, or being able to execute SPARQL queries on a single virtual RDF database that is distributed across more than one locally administered server [3][4][5]. The Bio2RDF server allows RDF based queries to be executed in parallel on multiple databases.

The Bio2RDF project also provides about 40 RDF databases, and the equivalent data-dumps, representing the major biological databases, together with SPARQL endpoints which can be used to query the RDF information contained in each database. The Bio2RDF server enables queries to be executed on not only these 40 databases, but also other related databases such as the Linked Open Drug Data, Neurocommons [4], and Dbpedia [6] databases, without requiring them to be loaded into a single database.

Some common queries have already been developed and published as part of the Bio2RDF server package, but user-specific queries can also be developed to answer specific questions. Figure 1 contains a brief overview of the process. These queries can be developed using a web interface that allows users to choose which endpoints to execute the query on, and which namespaces to use with the query. Users can execute the query with sample information and incrementally modify the query in order to obtain the desired results. The query and any associated namespaces and endpoints can be included in a personal installation of the Bio2RDF server by copying a URL into the server properties, or the resulting RDF configuration snippet into a local file.

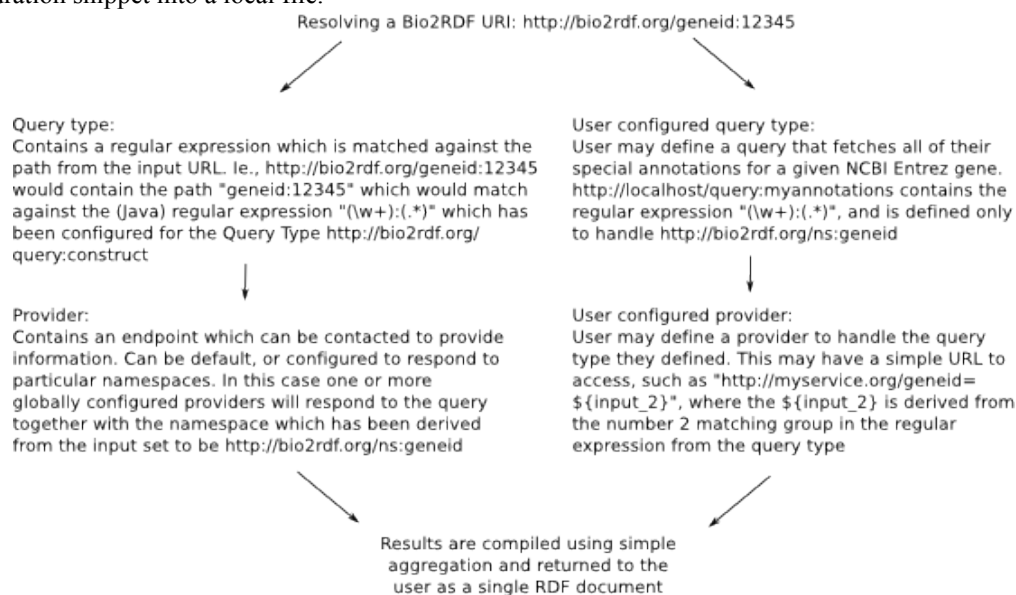


Figure 1: Resolving a Bio2RDF URI

As part of the process, users are required to login using an OpenID, and their contributions are attributed to this OpenID URI. This enables users to track their contributions, however, contributions are not currently able to be protected from changes on the community server unless they are included in the global configuration. Users can copy and paste the configuration to their local file system at any time in order to prevent changes occurring to it.

Users can also add new endpoints on which to either execute queries and have results returned to the Bio2RDF server in RDF, or redirect users to. A common use of the redirect mechanism is to transform the normalised Bio2RDF URI, ie., http://bio2rdf.org/my_namespace:private_identifier, into a URL that will contain an HTML representation of the object from the namespace “my_namespace” with the database specific identifier, “private_identifier.” The most common queries however perform SPARQL queries and return the resulting RDF as part of the resulting document to the user. These queries can be executed similarly across different endpoints because the RDF model follows a common format that does not require a user to know which properties or schemas are required by a particular database. This simplifies the otherwise mammoth requirement of constructing many similar queries across each database.

Users do not have to choose unique query URI's unless there is an actual semantic conflict. More than one query, and more than one endpoint, may be used to respond to a given query. For instance, there are a number of different queries that can be used, with corresponding endpoints, in resolving the normalised identifier, <http://bio2rdf.org/namespace:identifier>. Some of these queries depend on “namespace” matching a namespace that has been assigned to a particular provider, but others can be executed without recognising the namespace, as they are not semantically specific to the namespace, or the endpoint being used is a generic source of information.

Generic sources of information are known internally as default providers. They are used to substantially reduce the number of configuration elements required. Users can create their own default providers in order to avoid having to write similar configurations or to avoid assigning all of the currently known namespaces to their provider.

If the user cannot find a matching namespace in the current set of more than 1500 Bio2RDF assigned namespaces then they can also create a new namespace and assign it to their queries and provider endpoints. The namespace should be unique in order to keep a one to one relationship between the short namespace prefix and a particular part of a database that it refers to.

The web application was created in JSP, and it uses the existing prototype Bio2RDF server libraries which have been implemented using Java. The data is stored in an RDF database, which is accessed and updated using the extended SPARQL Update syntax. The database contains extensions to allow SPARQL Graph level security, so the web application cannot be used to edit information such as the global configuration which has been previously loaded into the database in a different graph with read only permissions.

REFERENCES

1. Lambrix, P. & Jakoniene, V.; *Towards transparent access to multiple biological databanks* Proceedings of the First Asia-Pacific bioinformatics conference on Bioinformatics, 2003 **19**: p. 53-60.
2. Belleau, F.; et al., *Bio2RDF: Towards a mashup to build bioinformatics knowledge systems*. Journal of Biomedical Informatics, 2008, **41**: p. 706-716.
3. Goble, C. A.; et al., *Transparent access to multiple bioinformatics information sources* IBM Systems Journal, 2001, **40**: p. 532-551.
4. Ruttenberg, A.; Rees, J.; Samwald, M. & Marshall, M.; *Life sciences on the Semantic Web: the Neurocommons and beyond* Briefings in Bioinformatics, 2009, **10**: p. 193.
5. Pasquier, C.; *Biological data integration using Semantic Web technologies* Biochimie, 2008, **90**: p. 584-594.
6. Bizer, C.; et. al., *DBpedia – A Crystallization Point for the Web of Data*. To appear in: Journal of Web Semantics (JWS), Special Issue on the Web of Data. Retrieved on 29-06-2009 from <http://www.wiwiss.fu-berlin.de/en/institute/pwo/bizer/research/publications/Bizer-et-al-DBpedia-CrystallizationPoint-JWS-Preprint.pdf>.