SPANG: a command-line client supporting query generation for distributed SPARQL endpoints

Hirokazu Chiba and Ikuo Uchiyama

National Institute for Basic Biology, National Institutes of Natural Sciences,
Nishigonaka 38, Myodaiji, Okazaki, Aichi, 444-8585 Japan
{chiba,uchiyama}@nibb.ac.jp

Abstract. An increasing number of biological databases have been made available in the form of RDF that are accessible through SPARQL endpoints. These endpoints offer a valuable opportunity to utilize the RDF datasets as integrative databases. However, writing a SPARQL query often becomes a burden for biologists; thus, an easy-to-use querying tool for the SPARQL endpoints is necessary. Here, we developed SPANG, a command-line SPARQL client supporting query generation. SPANG can dynamically generate typical SPARQL queries according to the command-line arguments. SPANG can also use SPARQL templates existing in local system or on the Web. Further, SPANG allows a user to combine multiple queries, each with distinct target endpoints, via Unix pipe. These features enable easy access to RDF datasets through SPARQL endpoints, and enhance the integrative analysis of various biological data distributed across the Web. SPANG is freely available from `http://purl.org/net/spang`.

Keywords: SPARQL client, distributed SPARQL endpoints, SPARQL template library

Because of the rapid progress in biotechnologies, various types of biological data have been rapidly accumulating. The Semantic Web technology has attracted attention as a promising approach for integrating such growing heterogeneous data; thus, an increasing number of biological databases have been made available in the form of RDF through SPARQL endpoints [1–4]. Although these SPARQL endpoints provide an opportunity to utilize the RDF datasets as integrative databases, using SPARQL language often requires cumbersome coding tasks, such as prefix declaration of URIs and writing common code patterns repeatedly. Automation of such troublesome tasks will help the researchers use the biological databases through SPARQL endpoints. Here, we developed a novel SPARQL client, SPANG, for reducing the burden of coding SPARQL and increasing the reusability of written SPARQL codes.

SPANG is a command-line client that can dynamically generate SPARQL queries according to the command-line arguments. Basically, a single SPANG command submits a query to an endpoint. SPANG has two modes of operations: (i) SPARQL shortcut mode, where typical query patterns are generated according to command-line options; and (ii) SPARQL template mode, where the specified SPARQL template and parameters in the command line are used to generate a runtime query. The specified SPARQL template can be either a local file or a file on the Web. A normal SPARQL query that does not include parameters can also be executed in the SPARQL template mode. SPANG has several other mechanisms to simplify the cumbersome tasks in SPARQL querying; (a) prefix declarations described in configuration files are used for runtime autocompletion; (b) nicknames for SPARQL endpoints defined in configuration files can be used in the command line; and (c) SPARQL template libraries in the local system can be looked up by specifying a template name in the command line. Although the distributed SPANG package provides predefined configuration files and a template library for general use, each user can extend the settings by user-defined files.

Whereas a single SPANG command can submits a query to a specific endpoint, SPANG also enables combinatorial execution of multiple queries, each with distinct target endpoint, by connecting respective SPANG processes via Unix pipe. This functionality is similar to that of the SPARQL 1.1 federated query using the SERVICE keyword [5]. The code of a federated query includes nested subqueries and tends to be complicated. Instead, SPANG realizes combinatorial execution of multiple queries by distinct Unix processes connected via pipe to transfer variable bindings. Such a modular structure provides several merits; the reduced complexity of each query makes its implementation and debugging easier; and its combinatorial use with other queries or with other Unix commands offers a wide range of application.

Thus, SPANG enables easy access to RDF datasets through SPARQL endpoints, and also facilitate combinatorial use of distributed databases across the Web. These functionalities will enhance the integrative analysis of various biological data toward knowledge discovery. As a practical application, we will show a set of example queries using the UniProt SPARQL endpoint [2] and the MBGD SPARQL endpoint [4] to transfer protein annotations from well characterized genomes to poorly characterized genomes through the MBGD ortholog group information as a hub.

References

1. Katayama, T., Wilkinson, M.D., Aoki-Kinoshita, K.F., Kawashima, S., Yamamoto, Y., et al.: BioHackathon series in 2011 and 2012: penetration of ontology and linked data in life science domains. J. Biomed. Semantics 5, 5 (2014)
2. UniProt Consortium: Activities at the universal protein resource (UniProt). Nucleic Acids Res. 42, D191–D198 (2014)
3. Jupp, S., Malone, J., Bolleman, J., Brandizi, M., Davies, M., et al.: The EBI RDF platform: linked open data for the life sciences. Bioinformatics 30 1338-1339 (2014)
4. Chiba, H., Nishide, H., Uchiyama, I.: Construction of an ortholog database using the semantic web technology for integrative analysis of genomic data. PLoS ONE 10, e0122802 (2015)
5. Prud'hommeaux, E., Buil-Aranda, C.: SPARQL 1.1 federated query. W3C Recommendation (21 March 2013), http://www.w3.org/TR/sparql11-federated-query/