Published online 1 June 2011 *Nucleic Acids Research, 2011, Vol. 39, Web Server issue* ***W533–W540***
doi:10.1093/nar/gkr353

Semantic-JSON: a lightweight web service interface for Semantic Web contents integrating multiple life science databases

Norio Kobayashi, Manabu Ishii, Satoshi Takahashi, Yoshiki Mochizuki, Akihiro Matsushima and Tetsuro Toyoda*

Bioinformatics And Systems Engineering division (BASE), RIKEN. 1-7-22 Suehiro, Tsurumi, Yokohama, Kanagawa, 230-0045 Japan

Received February 18, 2011; Revised April 15, 2011; Accepted April 26, 2011

ABSTRACT

Global cloud frameworks for bioinformatics research databases become huge and heterogeneous; solutions face various diametric challenges comprising cross-integration, retrieval, security and openness. To address this, as of March 2011 organizations including RIKEN published 192 mammalian, plant and protein life sciences databases having 8.2 million data records, integrated as Linked Open or Private Data (LOD/LPD) using SciNetS.org, the Scientists' Networking System. The huge quantity of linked data this database integration framework covers is based on the Semantic Web, where researchers collaborate by managing metadata across public and private databases in a secured data space. This outstripped the data query capacity of existing interface tools like SPARQL. Actual research also requires specialized tools for data analysis using raw original data. To solve these challenges, in December 2009 we developed the lightweight Semantic-JSON interface to access each fragment of linked and raw life sciences data securely under the control of programming languages popularly used by bioinformaticians such as Perl and Ruby. Researchers successfully used the interface across 28 million semantic relationships for biological applications including genome design, sequence processing, inference over phenotype databases, full-text search indexing and human-readable contents like ontology and LOD tree viewers. Semantic-JSON services of SciNetS.org are provided at http://semanticjson.org.

INTRODUCTION

Integrating numerous life sciences databases and their cross-sectional analysis are fundamental to achieving comprehensive understanding of life phenomena. Semantic Web (1) provides a framework for realizing global metadata repositories to describe the semantics of data on the World Wide Web. More specifically, this framework can integrate data over multiple databases published on different web servers and presented in a variety of heterogeneous data structures.

In the Semantic Web, each data item is named by a Unified Resource Identifier (URI) in order to refer to the data item on the Web, and a semantic relationship between two data items is written as triple of subject, predicate and object where subject and object correspond to data items and predicate corresponds to a semantic link named by a URI. A set of such triples is conventionally written in the standardized data format called Resource Description Framework (RDF) (http://www.w3.org/RDF/), forming a RDF graph as Linked Open Data (LOD) (2), and is accessed by the query language called SPARQL (http://www.w3.org/TR/rdf-sparql-query/). So far, the Universal Protein Resource (UniProt) (3), the Biological Pathway Exchange (BioPAX) (4) and MIRIAM Resources (5) have published Biological RDF data and SPARQL endpoints that serve SPARQL query services have also been published on the Web (6,7).

In practical usage of bioinformatic databases, researchers often need to integrate data processing of LOD and their Linked Private Data (LPD) which are unpublished linked data. Further, when generating a database, researchers first edit data using their private and public data in a securely closed data space under their collaboration, and then the database is published on the Web by giving access permission. The Semantic-Web based data integration and publishing framework called Scientist's

*To whom correspondence should be addressed. Tel: +81 45 503 9111; Fax: +81 45 503 9553; Email: toyoda@base.riken.jp

Networking System (SciNetS.org) (8) was developed in this manner, allowing users to securely edit LOD/LPD by setting data access control. As of March 2011, 8.2 million data items including LOD/LPD and biological raw data are integrated in SciNetS.org by giving standardized vocabulary such as ontology. Accessing such huge and secure linked data and analyzing life sciences raw data cannot be achieved by conventional query tools such as SPARQL, which would require a huge search space to perform queries and does not support an access-controlled secure search over the mixed web of LOD and LPD items including both original raw data and meta data for which each user's access must be regulated during their search. A new data access interface was required therefore for SciNetS.org.

We propose a novel approach called Semantic-JSON that realizes an application programming interface lightweight enough to access vast LOD/LPD and life science raw data as a data repository. By definition, the interface provides a Representational State Transfer (REST) Web service that retrieves Semantic Web data described in the lightweight JavaScript Object Notation (JSON) format (http://www.json.org/). The interface must support seamless search over a mixed web of LOD and LPD, to each item of which different authenticated users have different access rights.

This article describes the concept and specification with practical examples of Semantic-JSON, including examples to create comprehensive visual contents out of the Semantic Web contents.

MATERIALS AND METHODS

Concept of Semantic-JSON

The Semantic-JSON approach consists of two developmental issues: a secured, integrated and unified domain data repository and a lightweight data access interface.

Figure 1 shows the overall Semantic-JSON data repository concept. The repository realizes an integrated linked data set in a unified domain by collecting existing external RDF linked data in numerous external domains distributed on the Semantic Web. When external URI of an external domain exist, they are mapped to corresponding internal IDs, and these mapping relationships are managed in the Semantic-JSON repository as LOD. In the repository, each element of an RDF statement—namely the triple ⟨subject, property, object⟩—is labelled by an internal ID. The data item can be accessed via a global URI called the Semantic-JSON URI, which consists of an internal ID and a command name. Semantic-JSON defines a set of commands that obtains a small but relevant fragment of Semantic Web data, including data items and linked data described in JSON format.

A Semantic-JSON data repository forms a set of secured virtual database spaces called projects that manage RDF data including RDF classes, RDF individuals and RDF statements whose subjects are those classes or individuals. Data access permission is defined for each project; a project administrator can set open access permission for a public database or set access permission to only a closed user group for a private secured database completely invisible from a user who does not have data permission.

A typical semantic inference is performed in Semantic-JSON as follows (Figure 1): first, a user gives external URIs for starting data items. The Semantic-JSON repository converts these external URIs into corresponding internal IDs. The user traverses Semantic Web data by sending Semantic-JSON URIs with internal IDs in a series of requests. The resultant internal IDs are converted into corresponding external IDs using a mapping relationship described using LOD. This mechanism makes semantic inferences possible in numerous programming languages supporting JSON without using SPARQL.

When multiple Semantic-JSON repositories are made available on the Web, data items in different repositories can be associated with one another via common external URIs, even though their internal IDs are likely to be different. Integrated use over multiple repositories is, therefore, realized via LOD that describe mapping relationships between external and internal IDs. The syntax and semantics of Semantic-JSON URIs must be standardized, however, including a set of commands. In the next sections, we describe the Semantic-JSON infrastructure, specification and implementation.

Data preparation

In order to realize a single domain database repository, we employed the SciNetS.org, a framework for integrating databases based on Semantic Web technology. The system is designed to support secure academic information sharing, and to handle database sharing, collaboration and publication. A database developer on SciNetS.org can also design semantics with equivalent elements and methodology to RDF. The system assigns an internally unique ID to each data element in the system and its corresponding URI so that it forms a single domain database repository.

As of March 2011, SciNetS.org integrates 192 public databases comprising 8.2 million individual RDF data records, 28 million semantic relations and 120 million associated files including human-readable HTML contents and downloadable archive data files in Tab Separated Values (TSV) and RDF formats, with a size totalling 4.5 TB. Selected core categorically segmentalized integrated databases for bioinformatics (8) include the RIKEN Integrated Database of Mammals consisting of 79 omics living organism molecular databases for humans and mice (http://scinets.org/db/mammal), the RIKEN Integrated Database of Plants consisting of 30 omics databases for *Aradidopsis* (http://scinets.org/db/plant) and the RIKEN Integrated Database of Protein consisting of 18 databases related to protein structure (http://scinets.org/db/protein). Though these integrated databases are developed to publish databases generated by RIKEN, in order to add semantics to the data we imported popular public databases and ontologies by converting them into a Semantic Web by mapping external and internal IDs and generated semantic links for all related data pairs. The individual databases of each integrated database are

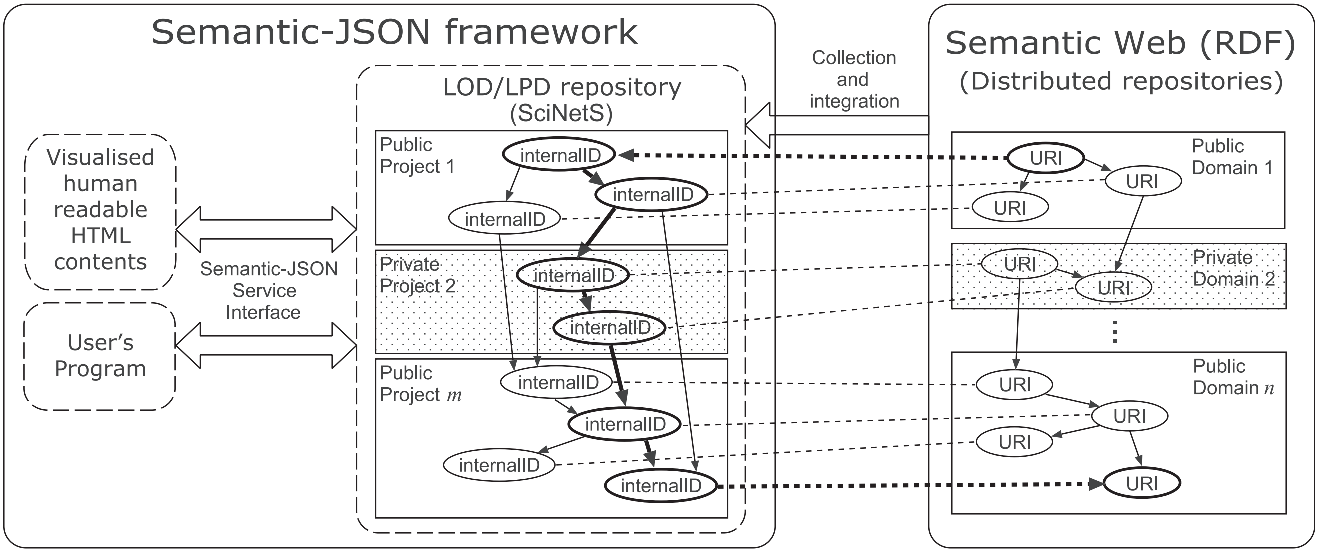


Figure 1. Concept of Semantic-JSON framework. Path of a semantic inference starts from a data item in Domain 1. First, a user resolves the internal ID associated with the external URI. Second, the user traverses data items on the Semantic-JSON LOD/LPD repository step by step using internal ID and internal URIs under data access controls. Third, the external ID defined in Domain *n* of the resultant internal ID is obtained.

listed in the Supplementary Data for this article. The data schema for each database and semantic links are defined by human curation, and imported to SciNetS by a program that follows the data schema. An LOD map of SciNets.org is shown in Figure 2.

DESIGN AND IMPLEMENTATION

Semantic-JSON URI

As we discussed, standardized syntax and semantics of a Semantic-JSON URI specifies a data record and also a command applied to the data record. The commands are classified into (i) link traverse commands that obtain data linked or reversely linked via a command-specific property from a data record and (ii) search commands. More concretely, (i) link traverse commands are defined as associated with standardized properties of RDF and RDF schema such as `label`, `type` and `subClassOf`. (ii) Search commands not only return data records that appear in an RDF statement whose subject, property or object include a user's query phrase that is often used for full-text search including `AND`, `OR` and `NOT` operations, but also return inferred data records that appear in an RDF graph sequentially connected by two RDF statements. Furthermore, also implemented for SciNetS.org are link traverse commands that employ data structure defined as an extension of RDF. Our Semantic-JSON tutorial (http://semantic-json.org) introduces these implemented commands.

Semantic-JSON URIs are specified as http or https requests that do not depend on the specific programming language used for implementation. A Semantic-JSON URI is written using the following syntax:

domain/json/*command*/[*lang*/]*internalID*[?*param*],

where *lang* denotes the natural language used for describing data content such as a language used to describe a data record label (*lang* has to be omitted if data processing command notation *command* does not depend on the language). The *param* notation is used to specify data resource and literal string search for user keywords. The *internalID* parameter uniquely identifies a data item in the repository located on the web server specified by *domain*. For instance, the URI that obtains an English label for the data item identified by URI http://semantic-json.org/item/rib158i of internal ID rib158i is written as

http://semantic-json.org/json/label/en/rib158i,

and the response in JSON format is

{‘label”:“RIKEN SciNetS\/SciNeS”, “lang”:“en”}

when the RDF statement ⟨rib158i, rdfs:label, “RIKEN SciNetS/SciNeS@en”⟩ exists.

Semantic-JSON Service

Semantic-JSON is implemented by a client–server model using HTTP or HTTPs protocol for communication on the Web. A server acting as a single domain data repository is developed as a REST web service provider that receives a Semantic-JSON URI as a request from an http client and returns the corresponding Semantic Web data presented in JSON format. We developed the Semantic-JSON server deployed at http://semantic-json.org by enabling access to integrated Semantic Web data introduced so far on SciNetS.org.

We developed a set of Semantic-JSON libraries in Ruby, Perl, Python, JavaScript, Java and Mathematica for the client programmer. We developed these libraries for existing http clients provided as a programming language library, and also developed JSON data processors to assist users to create Semantic-JSON URI and handle responding to the URI in JSON format as a native object in the corresponding programming language.

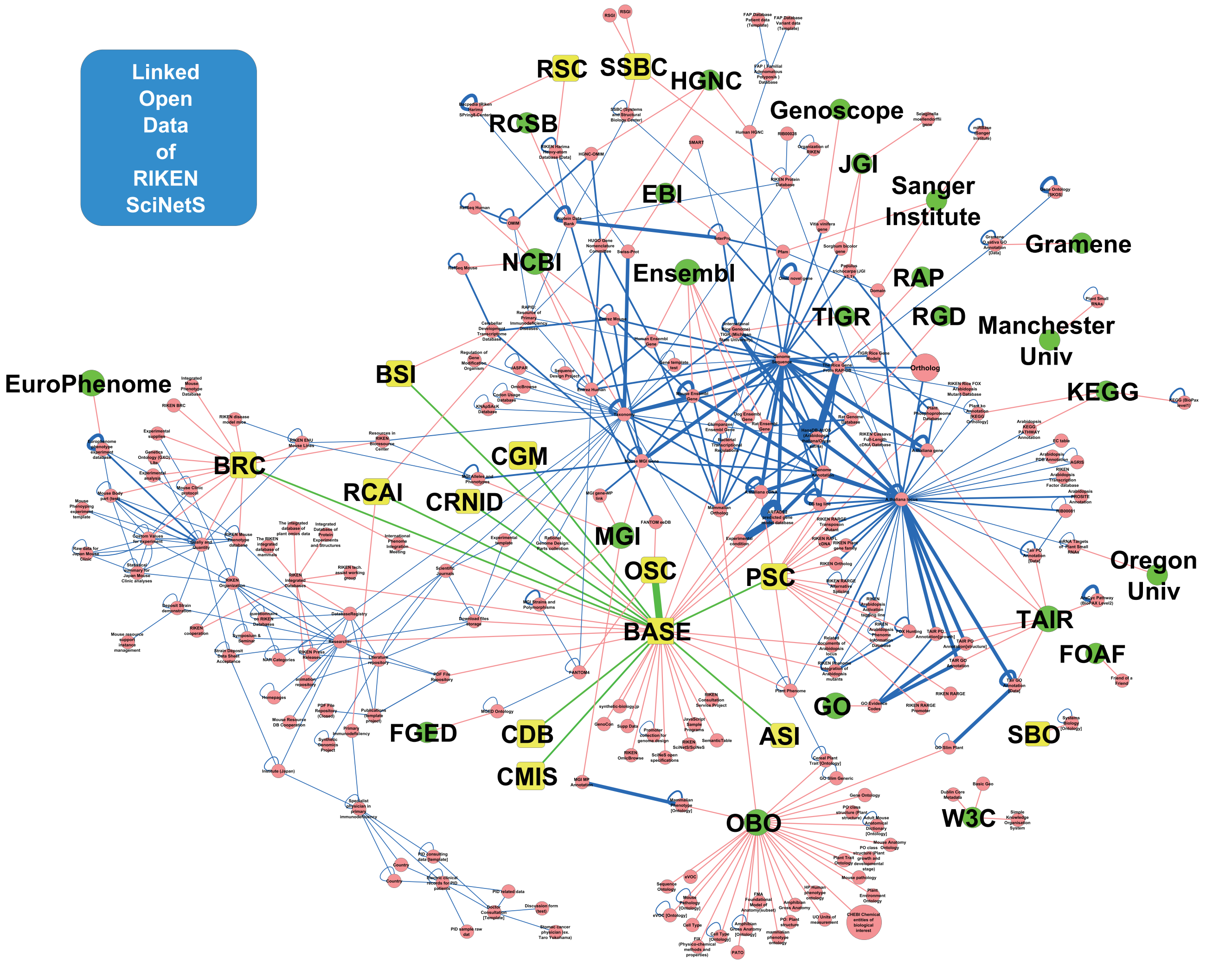


Figure 2. LOD map of selected databases integrated in the SciNetS.org unified domain data repository accessible via Semantic-JSON interface. Pink circles represent database projects, yellow squares represent RIKEN centers and green circles are external organizations. Dark blue links represent LOD between database projects, red links represent representative organization ownerships of database projects and light green links represent sharing of data items between RIKEN centers (http://scinets.org/item/crib140u2i).

Figure 3 shows a Perl program that obtains an English label as the example operation shown in the previous section. The Perl invoke function is also provided by the Semantic-JSON library for each other programming language as well. This encodes the Semantic-JSON server response from JSON format into a native object such as Hash and Array in the programming languages. A user can, therefore, write a program without considering JSON formatting.

Biological application

Here, we introduce biological application examples of Semantic-JSON client programs that access our Semantic-JSON server using the Semantic-JSON library, with the source codes for the examples and execution results introduced in the online Semantic-JSON tutorial.

Genome sequence processing. We start with a simple program to illustrate how a user programs a Semantic-JSON client. This program traverses RDF graphs shown in Figure 4 starting with an RDF individual of the *Arabidopsis* gene labeled 'ATCG00280' (9) and obtains the part of the genome sequence associated with the gene written in Fasta format.

The RDF individual of *Arabidopsis* gene which is the starting point of this example can be specified in several ways:

(1) If the external URI of starting data item as a Semantic Web item is known, its internal ID can be found by Semantic-JSON command internalID. (However, an external URI the RDF individual of *Arabidopsis* gene 'ATCG00280' shown in this sample is not defined.)
(2) Semantic-JSON command searchInstances can be used to find the data item by specifying a user's keyword 'ATCG00280'.
(3) Additionally, using SciNetS services provided at scinets.org, a user can find the starting data item by keyword search or by traversing dataset categories shown by the SciNetS.

```
# Load the Semantic-JSON library
use SemanticJson;

# Initialise a Client for Semantic-JSON service at http://semantic-json.org
my $client
      = SemanticJson->new("http://semantic-json.org");

# Obtain a JSON object having an English label of resource identified by 'rib158i'
$jObj = $client->invoke("label","en","rib158i",undef);

# Print the JSON object
print "name: ".$jObj->{"name"}."\n";
print "language: ".$jObj->{"lang"}."\n";
```

Figure 3. A fragment of a Perl program that obtains an English label of RDF Resource identified by internal ID 'rib158i' using the Semantic-JSON library. Function `invoke` is used to generate a Semantic-JSON URI using its argument's values, and returns the response in JSON format converted into Perl native objects such as Hash and Array.

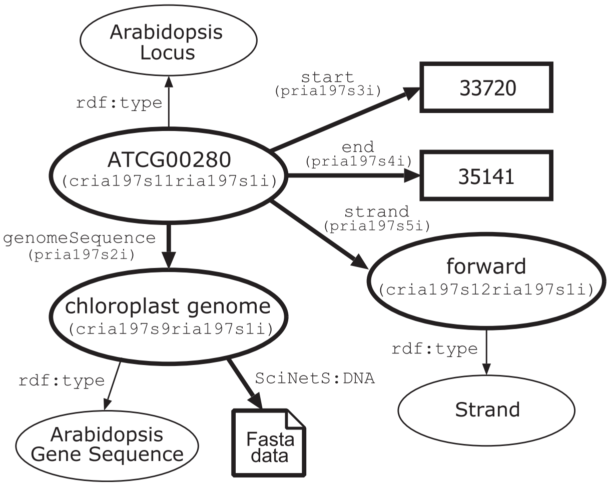


Figure 4. A fragment of RDF graph starting with *Arabidopsis* gene 'ATCG00280'. The gene links to its corresponding DNA genome sequence and genomic position data. Further, the genome sequence links to its sequence data described in the Fasta data.

The sample program finds statements whose subject is the *Arabidopsis* gene, and by invoking Semantic-JSON command `statements` obtains corresponding genome sequence and position data including start address, end address and strand. It then obtains in Fasta format a partial DNA sequence data of the genome sequence using the position data by employing the Semantic-JSON command `DNA` defined as an extension command for SciNetS.org. This simple example realizes comprehensive data processing over both RDF metadata and sequence data.

Inference over mouse phenotype databases. The program introduced here changes data schema by performing inferences on genotype information. A genotype is obtained from phenotype data annotation by following the semantic path of classes 'Annotation of Phenotype Data' → 'Attribute' → 'Entity' → 'Genotype' as shown in Figure 5. We define the novel property 'inferred genotype' that connects directly from individuals of 'Annotation of Phenotype Data' to an individual of 'Genotype' as a result of inference on the semantic path.

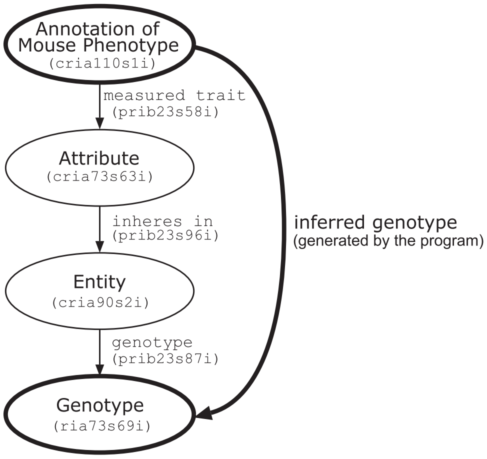


Figure 5. The data schema of the sequential semantic links from class 'Annotation of Mouse Phenotype' to class 'Genotype'.

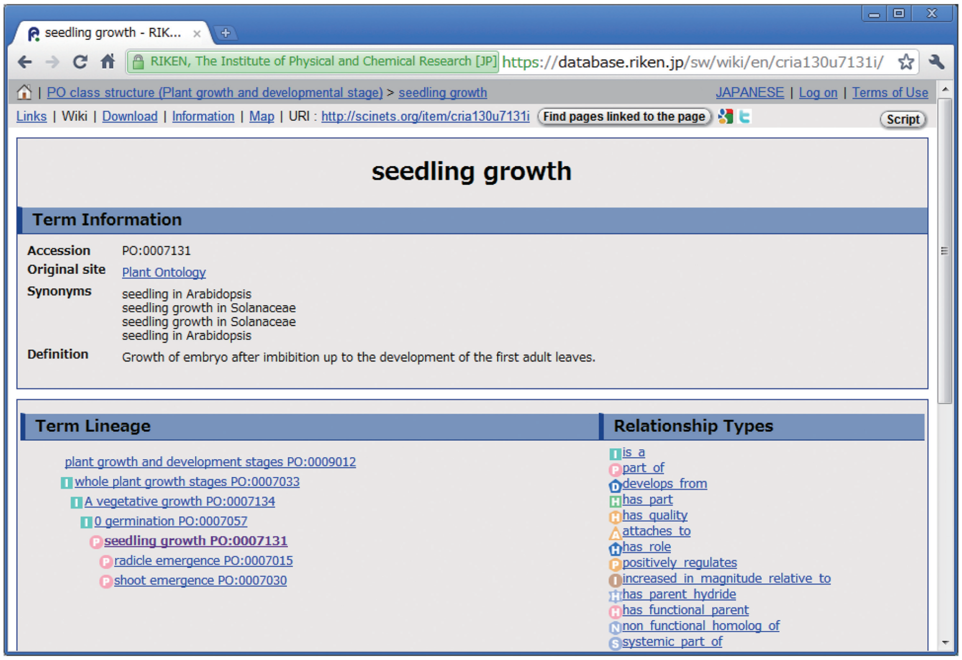
This is a typical problem that can be solved by applying graph pattern matching performed by SPARQL. Instead, by performing the command `statements`, the simple and straightforward algorithm in Semantic-JSON traverses RDF statements followed sequentially by the data schema. For practical pattern matching of huge graphs, Semantic-JSON allows users to control reducing the search space and the order of traversing the graph by introducing a pruning algorithm using Semantic-JSON search commands.

Ontology tree viewer. Ontologies developed for life sciences such as Gene Ontology (10) and Plant Ontology (11) are introduced in SciNetS.org. Using Semantic-JSON, SciNetS.org implements and deploys a graphical user interface as a web application that traverses ontology terms in a tree format.

Figure 6 is a snapshot of the ontology tree viewer page that shows the Plant Ontology (PO) term 'seedling growth' accessed by URL http://scinets.org/sw/wiki/en/cria130u7131i. In this tree view example, the starting node is the PO term 'seedling growth PO:0007131'. All ancestor nodes of the starting node are obtained by calling recursively the Semantic-JSON commands `superClasses` for a class hierarchy relationship and `reverseStatements` for other relationships. In the same way, child nodes of the start point are obtained by calling Semantic-JSON commands `subClasses` for a class hierarchy relationship and `statements` for other relationships.

LOD Tree Viewer. Semantic-JSON has been used to generate human-readable HTML content for a data item displaying LOD starting with the concerned data item in SciNetS.org.

Figure 7 is a part of the LOD view of a Ensembl human gene displayed in SciNetS.org. The content for the view is generated by a LOD crawler that traverses semantic links starting with a data item by calling Semantic-JSON commands implemented in Java.


seedling growth - RIK...
RIKEN, The Institute of Physical and Chemical Research [JP] https://database.riken.jp/sw/wiki/en/cria130u7131i/
| PO class structure (Plant growth and developmental stage) > seedling growth
JAPANESE | Log on | Terms of Use
Links | Wiki | Download | Information | Map | URI : http://scinets.org/item/cria130u7131i
Find pages linked to the page
Script

seedling growth

Term Information

Accession	PO:0007131
Original site	Plant Ontology
Synonyms	seedling in Arabidopsis seedling growth in Solanaceae seedling growth in Solanaceae seedling in Arabidopsis
Definition	Growth of embryo after imbibition up to the development of the first adult leaves.

Term Lineage

plant growth and development stages PO:0009012
I whole plant growth stages PO:0007033
I A vegetative growth PO:0007134
I 0 germination PO:0007057
P seedling growth PO:0007131
P radicle emergence PO:0007015
P shoot emergence PO:0007030

Relationship Types

I is_a
P part_of
D develops_from
H has_part
H has_quality
A attaches_to
H has_role
P positively_regulates
I increased_in_magnitude_relative_to
H has_parent_hydride
H has_functional_parent
N non_functional_homolog_of
S systemic_part_of

Figure 6. A snapshot of the PO ontology tree viewer embedded in SciNetS.org focusing on PO term 'seedling growth PO:0007131' (http://scinets.org/sw/wiki/en/cria130u7131i). The tree structure is programmably drawn using Semantic-JSON and published into a human-readable HTML content on SciNetS.org.

Other applications. Semantic-JSON is also used practically to develop a tool for genome design in *Arabidopsis* and to generate a full-text search index of the SciNetS.org repository for Semantic-JSON search commands.

The genome design tool includes a secured graphical user interface with a JavaScript program editor, its logger and its result display panel. The tool is embedded on SciNetS.org, so it is accessible via a user's Web browser. The tool was used for the 1st International Rational-Genome-Design Contest (GenoCon) in 2010 (http://genocon.org). The participants wrote programs in the secured personal programming environment of the genome design tool using it to obtain designed DNA sequences in *Arabidopsis* (14).

Semantic-JSON also has been used to generate and update a full-text search index by crawling all semantic graphs managed by SciNetS.org. Using the index, a Semantic-JSON search command is executed and as well a cross search on SciNetS.org (15).

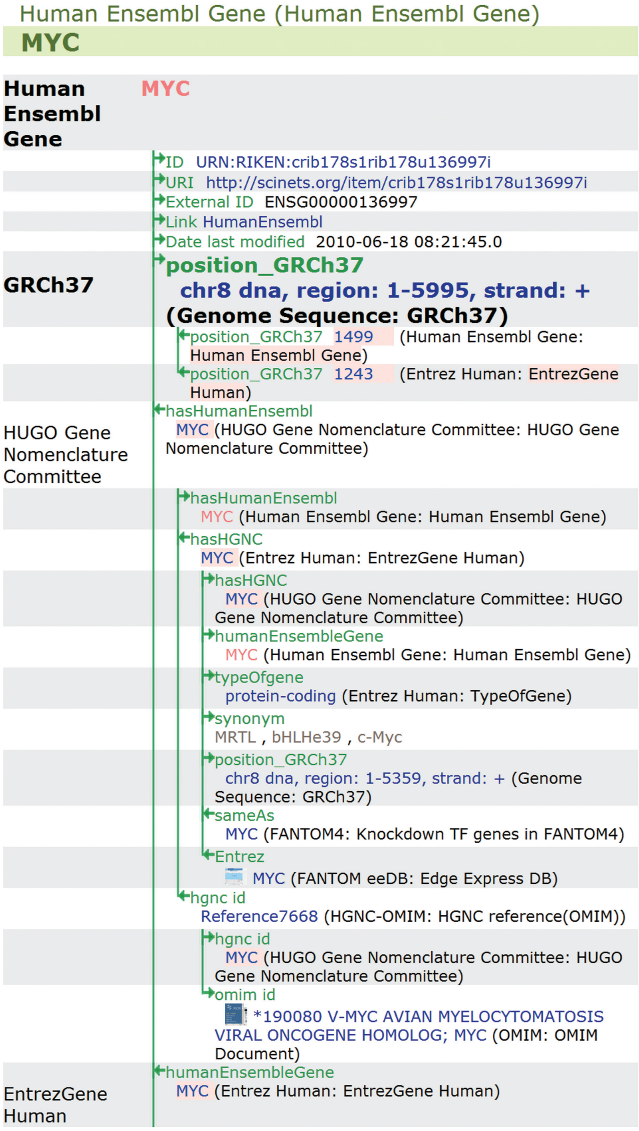
DISCUSSION AND CONCLUSIONS

As discussed so far, Semantic-JSON proposes a programming paradigm, different from standardized SPARQL, for Semantic Web data that realizes lightweight access to huge size LOD/LPD repositories. Since a Web browser can be used as a Semantic-JSON client, users can develop their program and check data by invoking Semantic-JSON URI via the Web browser. For example, since our Semantic-JSON server is integrated with SciNetS.org, users can also browse SciNetS.org data via the SciNetS.org graphic user interface on their own Web browser. On the negative side, Semantic-JSON supports primitive commands only; users need to implement their own custom data inference algorithm.

Since Semantic-JSON commands are specified as globally unique identifiers, namely URIs, the commands are performed as REST Web services. The Semantic JSON mechanism achieves traceability for execution of data analyzing programs that is requisite to realize genome design and a digital laboratory notebook that accesses Semantic Web resources in a reproducible fashion, where each step of the data analyzing processes to reach the program result is traceable. Auditability for safety over the design phase is indispensable to determine the category or class level of containment measures concerning genetic recombination experiments. The traceability and transparency of every single process in the complex design must be supported by the information platform (14).

Another advantage of Semantic-JSON is JSON support. Most programming languages support the JSON lightweight data container. SPARQL-JSON or a proposal for replying SPARQL results in JSON format (http://www.w3.org/TR?rdf-sparql-json-res/) would not address the data processing feature requirement or capacity demand satisfied by Semantic JSON for very large unified integrated data repositories such as SciNetS.org.

Short URL services (e.g. http://bit.ly) that are often used in social media such as Twitter (http://twitter.com) offer similar repository services. The services map external URLs under various domain names to internal URLs of a uniform domain using a short internal ID that corresponds the external one in the repository, as shown in Figure 1. Semantic-JSON expands on this idea by



Human Ensembl Gene (Human Ensembl Gene)

MYC

Human Ensembl Gene MYC

ID URN:RIKEN:crib178s1rib178u136997i
URI http://scinets.org/item/crib178s1rib178u136997i
External ID ENSG00000136997
Link HumanEnsembl
Date last modified 2010-06-18 08:21:45.0

GRCh37 position_GRCh37
chr8 dna, region: 1-5995, strand: +
(Genome Sequence: GRCh37)

position_GRCh37 1499 (Human Ensembl Gene: Human Ensembl Gene)
position_GRCh37 1243 (Entrez Human: EntrezGene Human)

hasHumanEnsembl
HUGO Gene Nomenclature Committee MYC (HUGO Gene Nomenclature Committee: HUGO Gene Nomenclature Committee)

hasHumanEnsembl
MYC (Human Ensembl Gene: Human Ensembl Gene)
hasHGNC
MYC (Entrez Human: EntrezGene Human)
hasHGNC
MYC (HUGO Gene Nomenclature Committee: HUGO Gene Nomenclature Committee)
humanEnsembleGene
MYC (Human Ensembl Gene: Human Ensembl Gene)
typeOfgene
protein-coding (Entrez Human: TypeOfGene)
synonym
MRTL , bHLHe39 , c-Myc
position_GRCh37
chr8 dna, region: 1-5359, strand: + (Genome Sequence: GRCh37)
sameAs
MYC (FANTOM4: Knockdown TF genes in FANTOM4)
Entrez
MYC (FANTOM eeDB: Edge Express DB)
hgnc id
Reference7668 (HGNC-OMIM: HGNC reference(OMIM))
hgnc id
MYC (HUGO Gene Nomenclature Committee: HUGO Gene Nomenclature Committee)
omim id
*190080 V-MYC AVIAN MYELOCYTOMATOSIS VIRAL ONCOGENE HOMOLOG; MYC (OMIM: OMIM Document)

humanEnsembleGene
EntrezGene Human MYC (Entrez Human: EntrezGene Human)

Figure 7. A part of the human readable HTML contents showing LOD starting with Ensembl human gene 'MYC' (http://scinets.org/item/crib178s1rib178u136997i). Green right and left arrows indicate forward and reverse semantic links, respectively, and follow their property names. The content includes LOD reached in traversing at most three steps of forward or reverse semantic links. The human gene is connected to data records of other databases including FANTOM 4 (12) and OMIM (13). In SciNetS.org, such human-readable content is generated for each RDF class and individuals by a data crawler implemented using Semantic-JSON.

designating that the repository have semantic relationships among the internal IDs that can be used to accelerate semantic communications in future social media.

In this article, we introduced the SciNetS.org single unified Semantic-JSON server system. Our goal is to develop Semantic Web cyber infrastructure that allows collaborative multiple distributed Semantic-JSON servers, as well as a Distributed Annotation System (DAS) (16) that implements distributed protocol to collaborate with multiple annotation servers or reference servers regarding genome data.

Since our Semantic-JSON server launched in December 2009, we have carefully tested and improved its response performance. Programs of users external to RIKEN have accessed our Semantic-JSON server ~134 000 times as of March, 2011.

Our future work includes extending commands to be used for life science data analysis and reduce data communication cost without losing lightweight Semantic Web data processing, and spread the beneficial use of the SciNetS.org community and its Semantic-JSON interface as a part of the life sciences Semantic Web data universe.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR online.

ACKNOWLEDGEMENTS

The authors would like to thank David Gifford for careful editing of the manuscript and providing graphic input and inspiring comments and suggestions.

FUNDING

Funding for open access charge: National Bioscience Database Center (NBDC) of Japan Science and Technology Agency (JST).

Conflict of interest statement. None declared.

REFERENCES

1. Berners-Lee,T., Hendler,J. and Lassila,O. (2001) The Semantic Web. *Scientific American*, **284**, 34–43.
2. Berners-Lee,T. (2006) Linked data. *Int. J. Semantic Web Inform. Syst.*, **4**, 2.
3. The UniProt Consortium. (2011) Ongoing and future developments at the Universal Protein Resource. *Nucleic Acids Res.*, **39**, D214–D219.
4. Demir,E., Cary,M.P., Paley,S., Fukuda,K., Lemer,C., Vastrik,I., Wu,G., D'Eustachio,P., Schaefer,C., Luciano,J. *et al.* (2010) The BioPAX community standard for pathway data sharing. *Nat. Biotechnol.*, **28**, 935–942.
5. Laibe,C. and Le Nove're,N. (2007) MIRIAM Resources: tools to generate and resolve robust cross-references in Systems Biology. *BMC Syst. Biol.*, **1**, 58.
6. Antezana,E., Blonde,W., Egana,M., Rutherford,A., Stevens,R., De Baets,B., Mironov,V. and Kuiper,M. (2009) BioGateway: a semantic systems biology tool for the life sciences. *BMC Bioinformatics*, **10(Suppl. 10)**, S11.
7. Belleau,F., Nolin,M.A., Tourigny,N., Rigault,P. and Morissette,J. (2008) Bio2RDF: towards a mashup to build bioinformatics knowledge systems. *J. Biomed. Inform.*, **41**, 706–716.
8. Masuya,H., Makita,Y., Kobayashi,N., Nishikata,K., Yoshida,Y., Mochizuki,Y., Doi,K., Takatsuki,T., Waki,K., Tanaka,N. *et al.* (2011) The RIKEN integrated database of mammals. *Nucleic Acids Res.*, **39**, D861–D870.
9. Swarbreck,D., Wilks,C., Lamesch,P., Berardini,T.Z., Garcia-Hernandez,M., Foerster,H., Li,D., Meyer,T., Muller,R., Ploetz,L. *et al.* (2008) The Arabidopsis Information Resource (TAIR): gene structure and function annotation. *Nucleic Acids Res.*, **36**, D1009–D1014.
10. Ashburner,M., Ball,C.A., Blake,J.A., Botstein,D., Butler,H., Cherry,J.M., Davis,A.P., Dolinski,K., Dwight,S.S., Eppig,J.T. *et al.* (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, **25**, 25–29.

11. Jaiswal,P., Avraham,S., Ilic,K., Kellogg,E., McCouch,S., Pujar,A., Reiser,L., Rhee,S.Y., Sachs,M.M., Schaeffer,M. *et al.* (2005) Plant Ontology (PO): a controlled vocabulary of plant structures and growth stages. *Comparative Funct. Genomics*, **6**, 388–397.
12. Kawaji,H., Severin,J., Lizio,M., Forrest,A.R., van Nimwegen,E., Rehli,M., Schroder,K., Irvine,K., Suzuki,H., Carninci,P. *et al.* (2011) Update of the FANTOM web resource: from mammalian transcriptional landscape to its dynamic regulation. *Nucleic Acids Res.*, **39**, D856–D860.
13. Hamosh,A., Scott,A.F., Amberger,J.S., Bocchini,C.A. and McKusick,V.A. (2005) Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Res.*, **33**, D514–D517.
14. Toyoda,T. (2011) Methods for open innovation on a genome-design platform associating scientific, commercial, and educational communitites in synthetic biology. *Methods Enzymol.*, **48**, 9.
15. Iida,K., Kawaguchi,S., Kobayashi,N., Yoshida,Y., Ishii,M., Harada,E., Hanada,K., Matsui,A., Okamoto,M., Ishida,J. *et al.* (2011) ARTADE2DB: Improved Statistical Inferences for Arabidopsis Gene Functions and Structure Predictions by Dynamic Structure-Based Dynamic Expression (DSDE) Analyses. *Plant Cell Physiol.*, **52**, 254–264.
16. Jenkinson,A.M., Albrecht,M., Birney,E., Blankenburg,H., Down,T., Finn,R.D., Hermjakob,H., Hubbard,T.J., Jimenez,R.C., Jones,P. *et al.* (2008) Integrating biological data – the distributed annotation system. *BMC Bioinformatics*, **9(Suppl. 8)**, S3.