

博 士 論 文

A study on standardization and
interoperability of biological databases

(生命科学データベースの標準化と
相互運用性に関する研究)

片山 俊明

A Dissertation Presented

by

Toshiaki Katayama

Submitted to

the Department of Computational Biology and Medical Sciences of
the Graduate School of Frontier Sciences of the University of Tokyo

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Acknowledgements

In collaboration with the Database Center for Life Science, I started the work described in this thesis when I was employed by the Human Genome Center of the University of Tokyo. I would like to express my deepest gratitude to Prof. Toshihisa Takagi for his continuous support on my work including the BioHackathon over the past ten years. I am truly grateful to Prof. Minoru Kanehisa for supervising my research on the KEGG database and its Web services. My sincere gratitude goes to Prof. Yutaka Suzuki, Prof. Kiyoshi Asai, Prof. Kouji Kozaki, and Prof. Kiyoko F. Aoki-Kinoshita who constituted the thesis committee with Prof. Toshihisa Takagi.

I am grateful to Dr. Mitsuteru Nakao and Dr. Naohisa Goto for developing and maintaining the BioRuby library for years and I thank Mr. Raoul Bonnal, Dr. Pjotr Prins, Dr. Jan Aerts, and all the contributors of the BioRuby library and the RubyGems packages. I am also grateful to Dr. Mitsuteru Nakao for designing the TogoWS APIs, Dr. Hiroyuki Mishima for the development of the Ruby UCSC API, Drs. Shuichi Kawashima, Shinobu Okamoto, Yuki Moriya, Hirokazu Chiba, Yuki Naito, Takatomo Fujisawa, and Hiroshi Mori for the fundamental design and the development of ontologies and RDF datasets for TogoGenome. I also gratefully acknowledge Mr. Tatsuya Nishizawa for his support in the development of the TogoWS service, Ms. Yoko Okabeppu, Mr. Akio Nagano, Mr. Daisuke Satoh, Mr. Keita Urashima, Mr. Yoji Shidara, Mr. Naoki Nishiguchi, and the engineers at Eiwa System Management Inc. for the software development and maintenance of the TogoGenome and TogoStanza systems. I am thankful to Ms. Nozomi Yamamoto, Ms. Hiroyo Nishide and the participants of the monthly SPARQLthon meetings for fruitful discussions, contributions, and development of the ontologies and TogoStanza in the multiomics domains.

I would also like to thank the participants of the BioHackathon series and the RDF summits for valuable discussions on Web Services and the Semantic Web. In particular, I would like to acknowledge Mr. Jerven Bolleman for inventing the FALDO ontology and

Mr. Robert Buels for his contributions to develop support for SPARQL in JBrowse.

For fulfilling requirements of the degree, I am really grateful to Prof. Koichi Ito, Prof. Shinichi Morishita and Prof. Masahiro Kasahara for their support and advice. I also thank all my colleagues and staff of the Bioinformatics Center of Kyoto University, the Human Genome Center of the University of Tokyo, the National Bioscience Database Center, and the Database Center for Life Science. Lastly, I thank my family for their kind support and understanding.

Table of Contents

Acknowledgements	1
Table of Contents	3
Chapter 1	5
Introduction	5
1.1 Background	5
1.2 Objectives	6
1.2.1 Standardization and interoperability of database access	6
1.2.2 Standardization and interoperability of database contents	7
Chapter 2	10
Standardization and interoperability of database access with Web Services	10
2.1 Introduction	10
2.2 TogoWS REST API	12
2.2.1 Database search	13
2.2.2 Hit count and pagination	14
2.2.3 Entry retrieval	14
2.2.4 Entry field extraction	17
2.2.5 Entry format conversion	18
2.2.6 Data format conversion	19
2.2.7 Performance tuning and error handling	20
2.3 TogoWS SOAP API	23
2.3.1 Integrated WSDL file	23
2.3.2 Sample code and documents	23
2.4 Server status monitor	24
2.5 Discussion	25
Chapter 3	26
Standardization and interoperability of database contents with Semantic Web technologies	26
3.1 Background	26
3.2 Results	28
3.2.1 TogoGenome	29

3.2.1.2 Semantic comparative genomics.....	32
3.2.2 TogoStanza	34
3.3 Methods.....	40
3.3.1 Integration of genome annotations.....	40
3.3.2 TogoGenome datasets.....	42
3.3.3 Development of TogoGenome.....	47
3.3.4 Development of TogoStanza.....	49
3.4 Discussion	50
3.5 Conclusions	51
Chapter 4	52
Discussions and conclusions	52
References	55
Supplemental Figures and Tables	58
Appendix.....	68
TogoWS API specification.....	68
TogoWS REST API conventions.....	68
Entry.....	68
Search.....	69
Convert.....	69
TogoWS external API	70
UCSC API	70
TogoWS API examples.....	72
TogoWS entry retrieval API examples	72
TogoWS search API examples	76
TogoWS convert API examples.....	78
TogoWS external API examples.....	80
TogoStanza examples	89
TogoStanza in a gene report page.....	89
TogoStanza in an organism report page.....	97
TogoStanza in an environment report page.....	103
TogoStanza in a phenotype report page.....	107

Chapter 1

Introduction

1.1 Background

In the life sciences domain, major biological databases such as protein tertiary structures, amino acid sequences, and nucleic acid sequences have already been established in the 1970s, and a culture to release research data for public use has been grown to maturity since then. This is the foundation for a wide range of research and development thereof from current basic biology to genome medical science.

As the international genome project progressed in the 1990s, information science supporting the construction of workflows for large-scale sequence analysis greatly advanced. At the same time, development and sharing of software that can be freely used in bioinformatics along with the open source movement including the GNU project and Linux have become popular.

Life science databases still continue to increase in quantity and variety, and there is increasing necessity to integrally use these enormous datasets. However, individual databases have different formats, IDs and vocabulary systems, and new concepts and data formats are being introduced along with new technologies.

For this reason, in bioinformatics research, the proportion of preprocessing such as data retrieval, conversion of data formats, resolving relationships between IDs and arrangement of meanings of data has increased, which was reported as a problem (NIH strategic plan for data science; <https://datascience.nih.gov/strategicplanrelease>) in 2018. To make this process efficient, it is necessary to standardize data and improve interoperability through international collaboration. In this research, in order to overcome

these problems, I developed Web services independent of the execution environment and constructed a genome database system integrating various data by Semantic Web technologies.

1.2 Objectives

1.2.1 Standardization and interoperability of database access

In order to build workflows of genome analysis, development of bioinformatics libraries for each programming language such as BioPerl (Stajich *et al.*, 2002), Biopython (Cock *et al.*, 2009), BioJava (Holland *et al.*, 2008), etc. as open source software has progressed since the beginning of the 2000s. Since I was working on the construction of the Kyoto Encyclopedia of Genes and Genomes (Kanehisa *et al.*, 2010), which is a database of genome and pathway information, I have been developing the BioRuby library (Goto *et al.*, 2010b) using the Ruby language in anticipation of data analysis in the post-genomic era. With the Ruby language, it was straightforward to achieve compatibility between modeling complex data such as object-oriented pathways and rapid program development which is a feature of scripting languages. On the other hand, in order to use libraries of various languages, it is necessary to install and code a program even for basic information processing such as data retrieval from databases, conversion of data formats, and construction of other workflows. Also, it took time and effort to build the environment to run on another computer. TogoWS (Katayama, Nakao, *et al.*, 2010a), developed in this research, eliminated the necessity of installation and dependency on any programming language and a computer environment by converting this functionality into Web services.

TogoWS supports major databases of the National Center for Biotechnology Information (NCBI), the European Bioinformatics Institute (EBI), the Protein Data Bank Japan (PDBj), the National Institute of Genetics DDBJ Center (DDBJ), the Kyoto Encyclopedia of Genes and Genomes (KEGG), and University of California Santa Cruz (UCSC). Because the methods for data retrieval provided by these centers were not unified, it was necessary for users to become familiar with their usage. Furthermore, the results obtained also varied, such as XML and original data format. In order to extract the necessary

information and construct a workflow, it is necessary to develop a program to parse the returned information for each data.

TogoWS provides APIs common to all databases for search, data retrieval, parse and conversion in order to treat them in a unified manner. For example, when acquiring a database entry, users can specify the database name and entry ID in the format of "/entry/DB/ID" following "http://togows.dbcls.jp". In addition, data is parsed by adding "/field_name" to extract subelements in the entry, and finally, the output format can be specified as ".xml" or ".json". These are realized by giving TogoWS server side the functions of BioPerl and BioRuby, so that users can obtain information without creating programs. In order to freely access vast genome annotation information including the human genome provided by UCSC, it was necessary to query UCSC's MySQL database in SQL. By adopting the Ruby UCSC API (Mishima *et al.*, 2012) in TogoWS, it became easy to access it just by its REST API. The service of TogoWS has been used stably from various bioinformatics applications for more than ten years and has contributed to standardization and interoperability of database access.

1.2.2 Standardization and interoperability of database contents

By having unified access to the major databases with TogoWS, it became easier to construct a workflow to acquire and process data, but for integrated use based on the meaning of data, it has become clear that it is necessary to standardize and enhance interoperability of the database contents themselves. For this reason, it was decided to organize a series of international developer conference BioHackathons, where major database developers gather, discuss and develop new database technologies. In these BioHackathons, which have been held for over ten years since 2008, adoption of Semantic Web technologies was proposed to improve standardization and interoperability of data (Katayama, Arakawa, *et al.*, 2010; Katayama *et al.*, 2011, 2013, 2014).

Semantic Web is a standard for constructing a Web of data proposed by Tim Berners-Lee who made the World Wide Web (WWW). In Semantic Web, we use Uniform Resource Identifiers (URIs) as a universal identifier that points to data. In addition, the meaning of

data and the relationship between data are expressed using standard vocabulary (ontology) defined by the Web Ontology Language (OWL). Furthermore, to model data, Resource Description Framework (RDF) is adopted, and the information is described by a combination of subject, predicate, and object (triple). Finally, SPARQL Protocol and RDF Query Language (SPARQL) is used for retrieval of RDF data. These are standardized by the WWW Consortium (W3C) and are the fundamental technology for providing a database that can be freely accessed on the Internet. Based on this, we standardized common URIs and ontologies in the life sciences and promoted the integration of various data by converting the contents of each database into RDF.

In this research, I constructed a new genome database TogoGenome by integrating data on biological species, genomes, genes, phenotypes, and environments by RDF because genome annotation requires information integration from diverse databases.

For this purpose, we first promoted the use of Identifiers.org as a standard URI by international collaboration, and developed the FALDO ontology (Bolleman *et al.*, 2016) for expressing the genomic coordinate system as the basis of annotation. Subsequently, we collaborated with international researchers to develop ontologies for semantically describing information in the International Nucleotide Sequence Database Collaboration (INSDC), an ontology for species taxonomy, an ontology of microbial phenotypes, an ontology of habitat environments etc. Based on these, we integrated RDF data mainly with RefSeq for genomic information and UniProt for protein annotation and then added phenotypic and environmental annotations.

Finally, I developed TogoStanza, which searches this information with the SPARQL language and visualizes the results for each biologically meaningful unit. In TogoGenome, optimal TogoStanza are combined depending on the context such as genes and environments and displayed to form a report page that summarizes relevant information. Also, because TogoStanza can be reused in other web-based databases, we can make development more efficient by mutual use of TogoStanza in multiple genomic databases such as MicrobeDB.jp, MGD, and CyanoBase which were developed at the same time

in Japan. Through the construction of TogoGenome, RDF conversion of major databases in the life sciences has progressed, and standardization of database contents and improvement of interoperability supporting future data science could be realized.

In the following chapters, I describe my research on “Standardization and interoperability of database access with Web services” which represents the integration of Web services based on my TogoWS paper (Katayama, Nakao, *et al.*, 2010b) and “Standardization and interoperability of database contents with Semantic Web technologies” which illustrates the integration of heterogeneous genomic data from my published work on TogoGenome/TogoStanza (Katayama *et al.*, 2019).

Chapter 2

Standardization and interoperability of database access with Web Services

Web services have become widely used in bioinformatics analysis, but there exist incompatibilities in interfaces and data types, which prevent users from making full use of a combination of these services. Therefore, I have developed the TogoWS service to provide an integrated interface with advanced features. In the TogoWS REST API, I introduce a unified access method for major database resources through intuitive URIs that can be used to search, retrieve, parse and convert the database entries. The TogoWS SOAP API resolves compatibility issues found on the server and client-side SOAP implementations. The TogoWS service is freely available from <http://togows.dbcls.jp/>.

2.1 Introduction

In the early 2000s, major bioinformatics centers have begun providing SOAP-based (<http://www.w3.org/2002/ws/>) Web services that enable users to use these database resources with client programs in an automated manner. These include the E-Utilities service (Sayers *et al.*, 2009) provided by the National Center for Biotechnology Information (NCBI), Web services provided by the European Bioinformatics Institute (EBI) (Labarga *et al.*, 2007; Pillai *et al.*, 2005), the Web API for Bioinformatics (WABI) from the DNA Data Bank of Japan (DDBJ) (Sugawara and Miyazaki, 2003; Miyazaki *et al.*, 2004; Sugawara *et al.*, 2008; Kwon *et al.*, 2009), the Protein Data Bank Japan's (PDBj) Web services (Standley *et al.*, 2008), and the KEGG API service from the Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa *et al.*, 2010). Thanks to these services, users can easily perform various bioinformatics tasks through their choice of client software and can reproduce each procedure as a workflow.

However, when it comes to using these services in combination, there are several limitations (Stockinger *et al.*, 2008) to their interoperability and technological implementation. 1) There are no common ontologies for operations and objects in these Web services, resulting in inconsistent naming conventions and data types. 2) This incompatibility of data types requires format conversion of objects to use the output of one service as the input to the next service. 3) There are several services that require specific SOAP features that are not always supported in the available SOAP libraries, even for major programming languages. 4) The client developer needs to be aware of fail-safe mechanisms, such as temporary downtime of the server or the network, as well as environmental restrictions such as the maximum size of exchanged data.

To overcome these limitations (especially for 1 and 2), the BioMoby project (Wilkinson and Links, 2002; Vandervalk *et al.*, 2009) was begun to provide a central registry of operations and objects used in public Web services, along with applicable ontologies. In this way, a number of BioMoby-compliant services were developed, and the BioMoby client can find the service that is appropriate for the given type of object. The main problem here is that most major bioinformatics service providers are not compatible with the BioMoby standard, possibly because it requires a considerable amount of server-side effort. Furthermore, it is also difficult to enforce a set of standard data formats for interoperability among these providers.

To help resolve these problems, I organized DBCLS BioHackathons in 2008 (Katayama, Arakawa, *et al.*, 2010) and 2009 (Katayama *et al.*, 2011), international workshops focusing on Web services, drawing participants from many backgrounds, including Web service providers, developers of the Open Bio* libraries (Stajich and Lapp, 2006) and client applications, and database creators in emerging fields like glycoinformatics and interactomics. One interesting topic in the BioHackathon was the attempt to resolve the current limitations in interoperability among existing Web services. For this purpose, a workflow was proposed that pipelines services provided by DDBJ, PDBj and KEGG to find homologs using BLAST and annotate them with structural and pathway information. When this workflow is run in the Taverna environment (Hull *et al.*, 2006), users again

encountered the essential need for data format conversion. The Open Bio* libraries, including BioPerl (Stajich *et al.*, 2002), BioRuby (Goto *et al.*, 2010b), Biopython (Cock *et al.*, 2009), and BioJava (Holland *et al.*, 2008), provide parsers for major database entry and software output formats such as the BLAST report. However, users are required to install these libraries and to write code to use their functionality.

Building upon discussions from the BioHackathon, I began to develop TogoWS, an integrated Web service ("togo" is a Japanese word for "integration") that provides uniform access to database resources, parsers for database entries, and converters among major data formats. Bioinformatics Web services can be categorized into data-retrieval services and analysis services. Although both types of services can be exposed using either the REST (Fielding, 2000) or the SOAP architecture, REST is better suited for data-retrieval services and SOAP is more suitable for analysis services because the former can be easily mapped to resource URIs and the latter usually requires a long execution time or complex parameters.

In a survey I conducted prior to implementation of TogoWS, I discovered that most existing Web services, such as NCBI's E-utilities and EBI's Dbfetch, are designed to search and retrieve database entries maintained at each institution. Therefore, in TogoWS, I designed a REST-based Web service for accessing database resources in a unified manner, with intuitive URI notation for searching, retrieving, parsing, and converting the database entries. Also, I developed a unified SOAP-based Web service in TogoWS that proxies analysis services provided by Japanese institutions to resolve several incompatibilities found in these services. Supplemental documents and source code in major programming languages (Perl, Ruby, Python, and Java) are also provided.

2.2 TogoWS REST API

The TogoWS REST service provides intuitive application programming interfaces (APIs) to search, retrieve, parse, and convert database entries. In the following sections, I will describe these interfaces and the internal architecture of the REST service.

(a) <http://togows.dbcls.jp/search/uniprot/lung+cancer>

```
Q7Z5Q7_HUMAN
Q5WPA9_PIG
Q6K043_HUMAN
Q56VW8_HUMAN
Q8TE03_HUMAN
B7ZW06_HUMAN
DLEC1_HUMAN
KKLC1_MACFA
C4QDY3_SCHMA
DLEC1_RAT
B1B5Y4_HUMAN
DLEC1_MOUSE
Q8IWW0_HUMAN
CASC1_HUMAN
```

(b) <http://togows.dbcls.jp/entry/kegg-compound/C07481/name>

```
Caffeine
```

(c) <http://togows.dbcls.jp/entry/kegg-compound/C07481/mass>

```
194.0804
```

(d) <http://togows.dbcls.jp/entry/kegg-compound/C07481/enzymes>

```
1.13.12.- 1.14.14.1 1.17.5.- 2.1.1.160
```

(e) <http://togows.dbcls.jp/entry/kegg-compound/C07481/enzymes.json>

```
[["1.13.12.-", "1.14.14.1", "1.17.5.-", "2.1.1.160"]]
```

Figure 2.1 Examples of the TogoWS URIs and their outputs.

2.2.1 Database search

TogoWS provides a uniform query interface for various databases. The result of the database search can be considered a resource that is relevant to the query string. Therefore, I determined to map each database name (DB) and query string (QUERY) to a URI by the following convention.

```
http://togows.dbcls.jp/search/DB/QUERY
```

A list of currently available databases can be obtained by accessing the following URI without a database name (**Supplemental Table 2.1**).

```
http://togows.dbcls.jp/search/
```

As an example, a search against the UniProt database using the phrase "lung cancer" can be represented as follows.

```
http://togows.dbcls.jp/search/uniprot/lung+cancer
```

The returned text contains matched entry IDs, one per line (**Figure 2.1a**). The QUERY

can be a simple keyword or a URI-encoded string containing a structured query with logical operations. The given query is translated by the TogoWS server and then sent to the corresponding service.

2.2.2 Hit count and pagination

A database search often returns a long list of hits. To make the TogoWS search service scalable, I introduced a method for counting and pagination. To count the number of hits, simply add "/count" to the end of the query URI.

```
http://togows.dbcls.jp/search/uniprot/lung+cancer/count
```

Then, the user can retrieve any subset of the hits by indicating OFFSET and LIMIT numbers in the following format.

```
http://togows.dbcls.jp/search/DB/QUERY/OFFSET,LIMIT
```

For example, to obtain 10 results starting from the 100th hit:

```
http://togows.dbcls.jp/search/uniprot/lung+cancer/100,10
```

The user can iterate over the OFFSET value, starting from 1 and incrementing it by LIMIT until all hits have been retrieved.

2.2.3 Entry retrieval

Each database entry can be identified by a database name and a unique identifier; therefore, it can be easily represented as a unique URI. In the TogoWS REST API, I mapped database names and entry IDs to URIs by the following convention.

```
http://togows.dbcls.jp/entry/DB/ENTRY_ID
```

where the "/entry" prefix indicates a REST action to retrieve the resource specified by

DB and ENTRY_ID, which represent the name of the database and the entry ID string, respectively.

For example, the URI to retrieve a KEGG GENES database entry "sec:YDR074W" can be represented as follows, and it will return the flatfile entry as a text string, without any decoration.

<http://togows.dbcls.jp/entry/kegg-genes/sce:YDR074W>

Multiple entries can be retrieved at once by concatenating entry IDs with commas. Therefore, PubMed entries "18077471" and "19151099" can be retrieved at a time by accessing the following URI.

<http://togows.dbcls.jp/entry/ncbi-pubmed/18077471,19151099>

A list of currently available databases can be obtained by accessing the following URI without a database name (**Supplemental Table 2.2**).

<http://togows.dbcls.jp/entry/>

To obtain actual database entries, TogoWS internally uses existing SOAP or REST interfaces provided by each database (**Figure 2.2**). Since the TogoWS acts as a proxy to various data sources, the user does not need to worry about the internals of the SOAP messages or complex CGI parameters that each database usually requires for access. The TogoWS server also caches the retrieved entries for a period of time to avoid overloading the original servers.

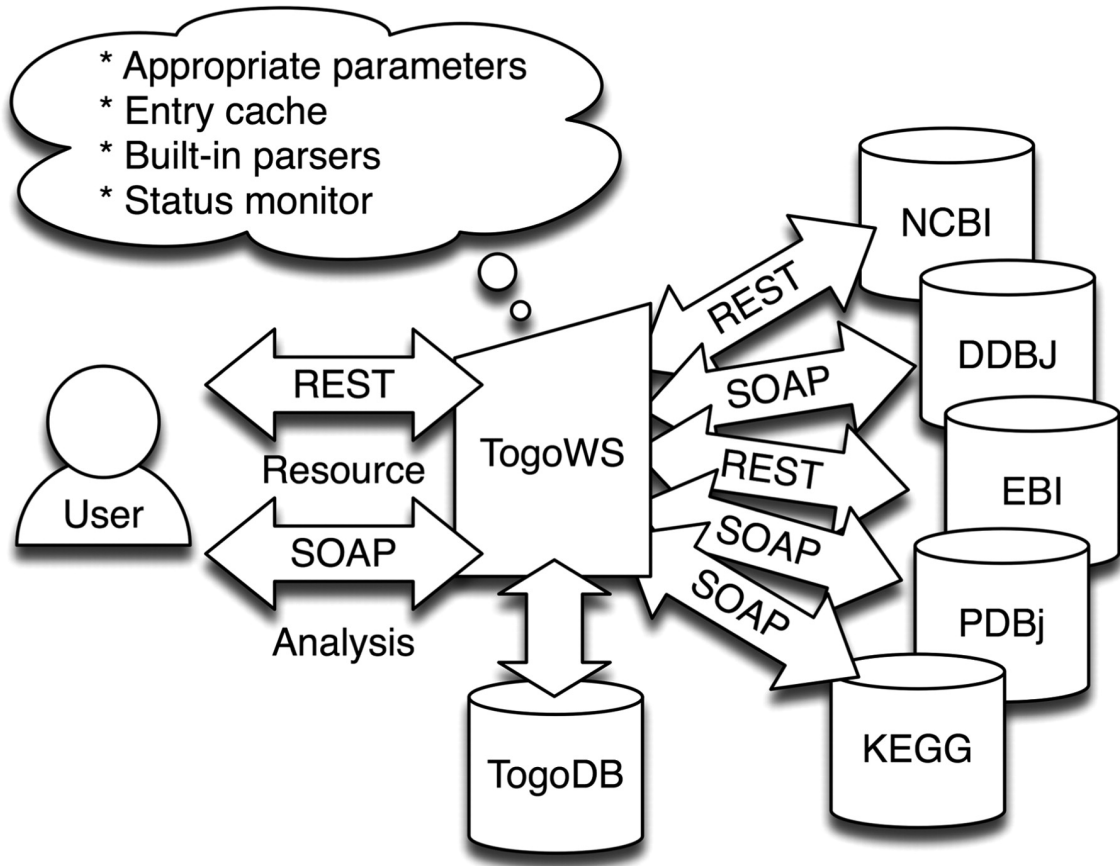


Figure 2.2 Schematic overview of the TogoWS service.

2.2.4 Entry field extraction

A unique feature of the TogoWS REST API is that it comes with built-in parsers for various database formats. Without this, the user will need to install a bioinformatics library such as BioPerl, Biopython, BioRuby, or BioJava and to write a program to extract the desired information from the retrieved entries. This requirement has been a bottleneck to the creation of an automated workflow that consumes a list of database entries and extracts information for the next step of the analysis pipeline. To resolve this situation, I embedded BioPerl and BioRuby libraries into the TogoWS server. These bioinformatics libraries cover a wide range of biomedical databases and provide efficient parsing functionality for various database entries. I extended the TogoWS REST API to support extraction of the field contents by adding a specific field name at the end of the URI, as follows

`http://togows.dbcls.jp/entry/DB/ENTRY_ID/FIELD`

where FIELD is one of the supported field names. The list of available field names differs from database to database and can be obtained by accessing the following URI.

`http://togows.dbcls.jp/entry/DB?fields`

As described in the previous section, TogoWS will retrieve specified entries from the original database. Then, the cached contents are internally processed by built-in parsers. In this manner, the user can access any field values of the given entries without programming.

For example, a name, a molecular weight, and relevant enzymes of the KEGG COMPOUND entry "C01083" can be extracted by the following URIs, respectively (**Figure 2.1b, 2.1c, 2.1d**).

`http://togows.dbcls.jp/entry/kegg-compound/C01083/name`

`http://togows.dbcls.jp/entry/kegg-compound/C01083/mass`

<http://togows.dbcls.jp/entry/kegg-compound/C01083/enzymes>

Similarly, the authors and abstract of the PubMed entry "19151099" can be retrieved by

<http://togows.dbcls.jp/entry/ncbi-pubmed/19151099/au>

<http://togows.dbcls.jp/entry/ncbi-pubmed/19151099/ab>

where "au" and "ab" correspond to the AU and AB lines, respectively, of the PubMed record in MEDLINE format.

2.2.5 Entry format conversion

Even though a specific field of an entry can be extracted, it is often required to convert the data format for further use. With the help of the built-in parsers described in the previous section, TogoWS provides format conversion of the entry simply by specifying the format as a URI suffix, analogous to the extension of a filename:

http://togows.dbcls.jp/entry/DB/ENTRY_ID.FORMAT

http://togows.dbcls.jp/entry/DB/ENTRY_ID/FIELD.FORMAT

For example, the DDBJ entry "M13899" can be converted to the FASTA, INSDC-XML, and GFF formats by the following URIs, respectively.

<http://togows.dbcls.jp/entry/ddbj/M13899.fasta>

<http://togows.dbcls.jp/entry/ddbj/M13899.xml>

<http://togows.dbcls.jp/entry/ddbj/M13899.gff>

Acceptable formats can vary according to the database and currently include XML, JSON, GFF version 3, FASTA, RDF/XML and Turtle. The FASTA and GFF formats are valid for nucleotide or peptide sequence databases, and the XML format is available if the original database is also provided as XML.

A list of available format names differs from database to database and can be obtained by

accessing the following URI.

<http://togows.dbcls.jp/entry/DB?formats>

Format conversion can also be applied to the extracted field. The following URI returns the associated enzymes of the KEGG COMPOUND entry "C01083" in JSON format (**Figure 2.1e**).

<http://togows.dbcls.jp/entry/kegg-compound/C01083/enzymes.json>

The JSON format (<https://tools.ietf.org/html/rfc4627>) is particularly useful when this service is used in a Web application that retrieves relevant information on-the-fly via an AJAX (<https://www.adaptivepath.org/ideas/ajax-new-approach-web-applications/>) method.

2.2.6 Data format conversion

TogoWS also provides format-to-format conversion functionality. Unlike the methods described above, this method uses the HTTP POST protocol instead of HTTP GET. The end-point URI of the data format conversion service uses the following convention.

<http://togows.dbcls.jp/convert/SOURCE.FORMAT>

For example, to convert a BLAST result to GFF format, simply POST the BLAST report string to the following URI.

<http://togows.dbcls.jp/convert/blast.gff>

The Ruby program (**Figure 2.3**) demonstrates how to read a BLAST output and convert its contents into GFF format.

Currently, GenBank, ENA, DDBJ, UniProt, BLAST, FASTA, GFF, GVF, PSL, Sim4,

HMMER, Exonerate, Wise, CSV, RDF/XML and Turtle formats are supported as source data types. This service is intended to be used in a workflow management software, in which the pipeline is often bottlenecked by incompatible data formats. TogoWS fills this kind of gap without requiring the user to install additional software on the local computer.

A list of currently available pairs of a source data type and a converted format can be obtained by accessing the following URI without a database name (**Supplemental Table 2.3**).

<http://togows.dbcls.jp/convert/>

2.2.7 Performance tuning and error handling

Because TogoWS relies on external Web services, it is important to reduce unnecessary accesses for these servers. Therefore, I introduced a cache mechanism which stores a retrieved database entry and reuses the data for future accesses to the same database entry. This cache system works efficiently in the case of entry field extraction described in Section 2.2.4, because the user often accesses the same entry by specifying different fields to be extracted. This mechanism also improves the response time of the TogoWS service especially for a large database entry such as an entire chromosome from the RefSeq database. Additionally, users can clear cached data by adding “?clear” to the entry retrieval URI as in the following format in case the cached content is out dated or broken.

http://togows.dbcls.jp/entry/DB/ENTRY_ID?clear

In order to avoid overload on external servers, it is important for TogoWS to comply with rules which are defined by these servers. For example, NCBI defines a rule that a client program must wait for several seconds between two or more successive accesses and the wait time can change depending on the time of day. TogoWS automatically applies this wait so that the user is not forced to write complex code to implement this wait logic in addition to retrieving data from these services.

Finally, TogoWS returns appropriate HTTP Error codes when the user's request is invalid or malformed (400 Bad Request), or the specified entry is not found (404 Not Found). These error codes are useful when writing a client program which retrieves a number of database entries at once and needs to capture the failure during the retrieval.

```

#!/usr/bin/env ruby

# Load libraries handling HTTP and CGI protocols and methods
require 'net/http'
require 'cgi'

# Read the BLAST output file
blast_output = File.read("blast_result.txt")

# Convert the output into a string suitable for a CGI query
post_data = CGI.escape(blast_output)

Net::HTTP.version_1_2

# Invoke HTTP connection to the TogoWS server
Net::HTTP.start('togows.dbcls.jp') { |http|
  # Execute the TogoWS conversion service via HTTP POST
  response = http.post('/convert/blast.gff', post_data)
  # Print out the result of conversion
  puts response.body
}

```

Figure 2.3 Example Ruby program to invoke the TogoWS conversion API for converting a BLAST output stored in the file "blast_result.txt" into GFF format.

2.3 TogoWS SOAP API

The other half of TogoWS is a SOAP-based proxy service for Japanese bioinformatics resources, which include DDBJ, PDBj and KEGG. In contrast to the REST service, SOAP is suitable for services requiring long execution time, returning structured objects, or expecting complex parameters in the query. The SOAP specification itself is an open standard and is independent of programming languages. However, its implementation in each programming language tends to be incomplete because of the complexity of the specification. Because of this, there appear to be several technical incompatibilities in each service. I have been collaboratively working with major institutions, including DDBJ and KEGG, to resolve the issues; however, there still remain problems that require modifications to their service specifications. These problems include the use of a MIME attachment for returning the results, the use of an HTTP cookie for stateful transactions, and different designs for asynchronous transactions, features that are not always supported by the SOAP library of choice.

2.3.1 Integrated WSDL file

Instead of asking all service providers to modify their services, I developed the TogoWS SOAP API, which proxies their services and thus hides the incompatibilities and differences between them. All services across these servers (DDBJ, PDBj, and KEGG) are integrated into only one WSDL file,

<http://togows.dbcls.jp/soap/wsdl/togows.wsdl>

so that the user can use all 368 operations that were originally spread among 26 WSDL files. This service has been tested in several major programming languages (Perl, Python, Ruby, and Java), so the user can use each service in the preferred language without difficulty. This approach also eliminates a burden on the service providers because they do not themselves need to test or improve the language compatibility of their services.

2.3.2 Sample code and documents

The TogoWS SOAP service comes with comprehensive sample code covering all

operations of the DDBJ, PDBj and KEGG services written in four programming languages (Perl, Python, Ruby, and Java). The user can freely examine and download the code from the following URL and use them as references for further development.

http://togodb.dbcls.jp/togows_domestic_method

Web services often lack documentation, forcing users to consult the WSDL file to learn what kind of operations are available, what data types are used for input and output, etc. However, this is not an effortless task, as the WSDL file was not designed to be read by a human. To remedy this problem, I have created a list of Web service operations from existing bioinformatics Web services worldwide.

http://togodb.dbcls.jp/togows_world_method

This list contains information extracted from the WSDL files, such as the description and input/output data types for 4,172 operations, including services integrated in the TogoWS SOAP API. In addition, I also assigned a functional classification to each operation.

2.4 Server status monitor

Web services are often used by computer programs in a pipeline. However, it is often difficult to detect temporary error caused by server-side problems. I have monitored the availability of all operations in DDBJ, PDBj, and KEGG over the past five years. The result is stored and summarized in the TogoWS status report.

<http://togows.dbcls.jp/monitor>

Since the monitoring is performed every day, these records may help the user determine whether the source of the problem is due to a local configuration or the remote server. The record also contains statistical information such as output size and response time, which has helped service providers to detect unexpected errors several times in the past.

2.5 Discussion

In TogoWS, I proposed an integrated service focused on the interface and compatibility of existing bioinformatics Web services. I successfully developed a REST interface for accessing database resources with intuitive and persistent URIs. This normalization of URIs was consequently found to be suitable as a method for generating unique resource URIs for making RDF data. For other services, I developed an integrated SOAP interface supplemented by sample code and a status monitor. However, I needed to terminate the SOAP interface in 2012 because most major bioinformatics centers discontinued their SOAP services. Instead, these centers began to replace their services with REST APIs. Thus, I have maintained TogoWS to conform with these changes and, subsequently, the REST interface of TogoWS has continued to be used for the past ten years from various bioinformatics applications.

Chapter 3

Standardization and interoperability of database contents with Semantic Web technologies

TogoGenome is a genome database that is purely based on Semantic Web technologies, which enables the integration of heterogeneous data and flexible semantic searches. All the information is stored as Resource Description Framework (RDF) data, and the reporting web pages are generated on the fly using SPARQL Protocol and RDF Query Language (SPARQL) queries. TogoGenome provides a semantic faceted search system by gene functional annotation, taxonomy, phenotypes, and environment based on the relevant ontologies. TogoGenome also serves as an interface to conduct semantic comparative genomics by which users can observe pan-organism or organism-specific genes based on the functional aspect of gene annotations and the combinations of organisms from different taxa. The TogoGenome database exhibits a modularized structure, and each module in the report pages is separately served as TogoStanza, which is a generic framework for rendering an information block as IFRAME/Web Components, which can, unlike several other monolithic databases, also be reused to construct other databases. TogoGenome and TogoStanza have been under development since 2012 and are freely available along with their source codes on the GitHub repositories at <https://github.com/togogenome/> and <https://github.com/togostanza/>, respectively, under the MIT license. Database URLs: <http://togogenome.org/> and <http://togostanza.org/>.

3.1 Background

In the life sciences, genome sequences have served as a central resource like a base map by which essential information, such as gene structures, regulatory regions, variations, and their functional annotations, could be integrated. As genome projects are conducted on various species, the genomic sequences and gene annotations are deposited into the International Nucleotide Sequence Database Collaboration (INSDC) (Cochrane *et al.*,

2016), which is jointly operated by the DNA Databank of Japan (DDBJ) (Mashima *et al.*, 2017), GenBank at NCBI (Benson *et al.*, 2017) and ENA at EMBL-EBI (Silvester *et al.*, 2018). However, each genome project often constructs its own genome database to add and update detailed annotations. For this purpose, generic and open source genome database systems such as GMOD (O'Connor *et al.*, 2008), Ensembl (Zerbino *et al.*, 2018) and InterMine (Kalderimis *et al.*, 2014) can be used.

These major database systems serve genome annotations for a large number of species. However, because these genome databases have been monolithically constructed, it is difficult to reuse their components even though the represented information is very similar. Meanwhile, to extend a system that represents information unique to an organism, the inclusion of additional annotations may require the modification of the database schema and further significant modifications to the system. In contrast, using RDF, because any data can be expressed in the same format, it is possible to easily integrate a wide variety of data from gene annotations to phenotypes and habitat environments of organisms. Also, there is no limit to the type of data that can be stored in an RDF database. Each piece of information integrated into the RDF database is distinguished by a globally unique identifier in the form of Uniform Resource Identifier (URI); thus the related information can be seamlessly linked and traced by the URIs.

Based on my experiences in the genome annotation and the construction of genome databases, I realized that most of the annotation information can be stylized. Therefore, it would be efficient to freely select the predefined modularized components for creating a genome database instance along with developing only new components based on annotations that are unique to the target organism stored as RDF data. Thus, it is expected that the cost required to construct a new genome database could be considerably reduced by managing all the annotation information in RDF and by providing the visualization modules for each subset of categorized annotations as reusable components. However, there was no precedent genome database that was purely based on RDF data; therefore, a demonstration was required to verify whether the use of SPARQL would be practical and scalable enough for a genome database.

3.2 Results

I developed a purely RDF-based genome database, TogoGenome, that was primarily based on the RefSeq (O’Leary *et al.*, 2016) and the UniProt (The UniProt Consortium, 2017) data. UniProt has been publishing their data in RDF since 2008 (UniProt Consortium, 2008), however, there has been no RDF representation of the RefSeq genome annotations. Therefore, in collaboration with DDBJ, I developed a converter of INSDC (DDBJ/GenBank/ENA) and RefSeq entries into RDF data. With a member of DDBJ, I also developed ontologies for the INSDC annotated sequences and taxonomy (<http://ddbj.nig.ac.jp/ontologies/>) as well as feature locations (see the **3.3.2 TogoGenome datasets** section).

TogoGenome uses the Semantic Web technology for data integration by which all the data are aggregated in RDF and semantically annotated with ontologies. Therefore, in addition to basic keyword searches, faceted searches with various aspects based on the semantic hierarchy of the data can be performed. Further, all the RDF data can be freely accessed by SPARQL queries not only from a web interface but also from a program. Bioinformaticians can easily develop a program to acquire the target datasets, perform analyses, and develop their own summaries and visualizations according to their requirements.

To produce reusable components, I developed TogoStanza, which is a framework for visualizing the result of a SPARQL query as an IFRAME or as Web Components (<https://www.webcomponents.org/>), which can be embedded into any HTML web page. Any number of components can be freely chosen and combined to generate a resulting page, which could not have been easily realized using the monolithic databases. In fact, TogoGenome displays various search results as a report page by combining with the related TogoStanza in an arbitrary context such as a gene, an organism, a phenotype, or an environment.

3.2.1 TogoGenome

TogoGenome is a Semantic Web-based genome database in which heterogeneous information is compiled from various RDF data annotated with ontologies. With the RDF data and ontologies, TogoGenome provides several query interfaces. First, users can conduct a faceted search based on a combination of gene, taxonomy, phenotype, and environment ontologies. Second, a simple comparative genomic analysis can be performed among the genes of several species based on the common and unique gene annotations. Finally, as in the traditional genome databases, TogoGenome data can be searched using a free text keyword or a genomic sequence. However, dedicated text indexing systems are required because SPARQL queries are not efficient enough to perform free text search (see the **3.3.3 Development of TogoGenome** section).

3.2.1.1 Ontology-based faceted search

One of the main interests of current biology is the relationships between genotypes and phenotypes. In the case of humans, the most important relation is between genes and diseases. In the case of crops and livestock, the genetic factors related to the aspects of quantity and quality, such as yield, nutritional value, and taste, are of considerable interest. In microorganisms, the effects of gene functions on physiology, metabolites, and interaction with the environment are typical examples of the subjects of research.

To elucidate these relations, a bioinformatics approach is required to efficiently formulate hypotheses using the knowledge in the databases and to verify these hypotheses by performing experiments. However, the genomic and phenotypic information are scattered throughout the genes, pathways, literature databases, and so on. There is no efficient database system to search for genes of various species in combination with the phenotypes.

As an example, suppose a scientist was attempting to verify the difference in the composition of cyanobacterial gene sets related to environmental responses, such as "histidine kinases," by comparing the gene sets of marine and freshwater living species.

This scientist must narrow down those genes that have the desired function by (1) obtaining a list of cyanobacteria from a taxonomy database, (2) selecting those species for which the complete genome has been decoded by searching genome databases, (3) identifying the growth environment of each cyanobacterium (“seawater” or “freshwater”) using the literature and other databases, (4) acquiring the gene set of each species, and (5) obtaining annotations for each gene set with the help of a gene ontology to acquire the intended gene set. This procedure is difficult to automate; therefore, it was necessary for researchers to manually investigate each database.

TogoGenome provides an ontology-based faceted search interface to easily obtain such information. Users can select “Cyanobacteria” from “Taxonomy,” specify “protein histidine kinase activity” from “GO: Molecular Function,” and select “saline water” and “fresh water” from “Environment” to obtain the desired gene sets (**Figure 3.1**).

TogoGenome faceted search

Comparison of "histidine kinase" genes of "saline water" and "fresh water" living "Cyanobacteria"

The figure displays two screenshots of the TogoGenome faceted search interface, comparing search results for histidine kinase genes in saline water versus fresh water environments.

Left Screenshot (Saline Water):

- GO: BiologicalProcess:** cellular nitrogen compound metabolic process
- GO: MolecularFunction:** protein histidine kinase activity
- GO: CellularComponent:** cytoplasm
- Taxonomy:** Cyanobacteria
- Phenotype:** Motile
- Environment:** saline water

Right Screenshot (Fresh Water):

- GO: BiologicalProcess:** cellular nitrogen compound metabolic process
- GO: MolecularFunction:** protein histidine kinase activity
- GO: CellularComponent:** cytoplasm
- Taxonomy:** Cyanobacteria
- Phenotype:** Motile
- Environment:** fresh water

Search Results (Left - Saline Water):

Gene	UniProt	Description	Gene ontology	Organism
167539:Pho_1121	Q7BHW3	Adaptive-response sensory-kinase SasA	phosphorelay sensor kinase activity, intracellular, ATP binding, rhythmic process	Prochlorococcus marinus subsp. marinus str. CCMP1375
74547:AKG35_RS05900	Q7V9P7	Adaptive-response sensory-kinase SasA	phosphorelay sensor kinase activity, intracellular, ATP binding, rhythmic process	Prochlorococcus marinus str. MIT 9313
59919:TX30_RS00785	Q7V113	Adaptive-response sensory-kinase SasA	phosphorelay sensor kinase activity, intracellular, ATP binding, rhythmic process	Prochlorococcus marinus subsp. pastoris str. CCMP1986
74546:PMT9312_RS02605	Q31AE8	Adaptive-response sensory-kinase SasA	phosphorelay sensor kinase activity, intracellular, rhythmic process	Prochlorococcus marinus str. MIT 9312
203124:TERY_RS21480	Q10WED	Histidine kinase	phosphorelay sensor kinase activity, intracellular, regulation of transcription, DNA-templated, ATP binding	Trichodesmium erythraeum IMS101
203124:TERY_RS04425	Q1178	Histidine kinase	phosphorelay sensor kinase activity, intracellular, integral component of membrane	Trichodesmium erythraeum IMS101
203124:TERY_RS19460	Q10XG2	Adaptive-response sensory-kinase SasA	phosphorelay sensor kinase activity, intracellular, circadian rhythm	Trichodesmium erythraeum IMS101
203124:TERY_RS13080	Q111B1	Histidine kinase	phosphorelay sensor kinase activity, intracellular	Trichodesmium erythraeum IMS101

Search Results (Right - Fresh Water):

Gene	UniProt	Description	Gene ontology	Organism
103690:PCC7120DELTA_RS11355	QBYV90	Histidine kinase	phosphorelay sensor kinase activity, intracellular, regulation of transcription, DNA-templated, ATP binding	Nostoc sp. PCC 7120
103690:PCC7120DELTA_RS16130	QBY151	Histidine kinase	phosphorelay sensor kinase activity, intracellular, regulation of transcription, DNA-templated, ATP binding, integral component of membrane	Nostoc sp. PCC 7120
103690:PCC7120DELTA_RS17535	Q9LCC2	Cyanobacterial phytochrome A	phosphorelay sensor kinase activity, intracellular, transcription, DNA-templated, regulation of transcription, DNA-templated, ATP binding, detection of visible light, photoreceptor activity, protein-chromophore linkage	Nostoc sp. PCC 7120
103690:PCC7120DELTA_RS12310	QBYV87	Histidine kinase	phosphorelay sensor kinase activity, intracellular, regulation of transcription, DNA-templated, ATP binding	Nostoc sp. PCC 7120
103690:PCC7120DELTA_RS08230	QBYXD4	Histidine kinase	phosphorelay sensor kinase activity, intracellular, regulation of transcription, DNA-templated, ATP binding	Nostoc sp. PCC 7120
103690:PCC7120DELTA_RS21635	QBYQ50	Histidine kinase	phosphorelay sensor kinase activity	Nostoc sp. PCC 7120

Figure 3.1 TogoGenome faceted search.

3.2.1.2 Semantic comparative genomics

Because UniProt proteins are semantically annotated in RDF and because TogoGenome holds the links between proteins and genes that are encoded in the genome of each organism, UniProt annotations can be used to find a specific subset of genes by selecting the attributes that are common or unique to a given set of species. First, the category of the annotation can be selected from: protein motif, sub-cellular location, pathway, gene ontology, enzyme classification, and ortholog classifications. Second, a maximum of five species can be selected to compare gene sets. Third, a list of functional classifications that are common only to the selected combination of organisms is presented. Fourth, one of the objective classifications can be selected to obtain a corresponding list of genes in the target organisms.

As an example, users can find genes having protein motifs unique to vertebrates by performing the following steps. (1) Select “Pfam motifs” as an annotation and (2) specify human, mouse, zebrafish, and sea squirt as the target set of organisms to perform the comparison. Further, users can (3a) select the combination of human \cap mouse \cap zebrafish and (4a) find the MHC domains corresponding to the adaptive immune system that are observed in vertebrates (therefore, not observed in sea squirts) (Dehal *et al.*, 2002), or (3b) select only the sea squirt and (4b) find Vanavin-2, which is a domain that is unique to sea squirts for oxygen binding (**Figure 3.2**).

Comparative genome

Step 1
Select an annotation aspect (e.g. Pfam motifs)

Step 2
Select organisms to compare (e.g. Human, Mouse, Zebrafish, and Sea squirt)

Step 3
Select combination of organisms (e.g. Human n Mouse n Zebrafish, or Sea squirt)

Step 4 & 5
Select an annotation and obtain genes (e.g. Vanabin)

Step 4 & 5
Select an annotation and obtain genes (e.g. MHC_I)

Step 4 & 5
Select an annotation and obtain genes (e.g. MHC_I)

Figure 3.2 TogoGenome comparative genomics.

3.2.1.3 Text index search

TogoGenome also provides simple keyword and sequence search interfaces. Because the text search function that is implemented in the existing RDF database is inadequate, TogoGenome uses Apache Solr (<http://lucene.apache.org/solr/>) to perform keyword search and the GGenome service to perform sequence search (see the Methods section). While performing keyword search, a list of TogoStanza, which contain the keywords, are presented on the basis of a free text match for gene names, species names, phenotype terms, and environmental terms. In a sequence search, a list of reference genomes, which includes a specified sequence, are exhibited with links to the TogoGenome genes, which reside in the overlapping or surrounding regions of the query sequence in the genome.

3.2.2 TogoStanza

The majority of the existing genome databases comprise typical components such as gene name and aliases with a brief description, chromosomal location and gene structures of the transcripts in a genome browser, the corresponding nucleotide sequences and amino acid sequences, the functional annotations of the genes and proteins, sequence variations and modifications, the corresponding ortholog genes in other species, relevant literature, and cross-references to external databases. Despite the fact that several pieces of information are commonly represented, they cannot be reused while developing a new database because most of the existing databases are monolithic. In fact, when my collaborators started to develop the MicrobeDB.jp (<https://microbedb.jp/>) and CyanoBase (Fujisawa *et al.*, 2017) databases, combining their original annotations with existing information that was stored in the major genome databases was difficult; therefore, they were forced to develop their own genome databases from scratch even though some of the contents were imported from the existing databases. To overcome this limitation, we developed TogoStanza to enable database developers to reuse components of the TogoGenome database in their genome databases. Because the TogoStanza system is designed to be generic, it is not limited to genome databases and is being utilized in other domains, such as proteomics and glycomics databases, as well as some other web applications.

3.2.2.1 Features of TogoStanza

TogoStanza is a web application framework that obtains information from a web API, SPARQL in particular, and visualizes the results as an IFRAME or Web Components that can be embedded into any web page (**Figure 3.3**). TogoGenome provides the report pages for each gene, organism, phenotype, and environment. The pages display all the information by combining a series of related TogoStanza. In the case of the gene report page, each TogoStanza takes a taxonomy ID and gene ID as its arguments, obtains information related to the gene using dedicated SPARQL queries, and visualizes the results in HTML. All technologies, such as HTTP, AJAX, HTML, CSS, and JavaScript, are web standards so that any web application developer can easily create or customize a TogoStanza for publication online, even though optimizing the performance of a SPARQL query may require some specialized tuning techniques based on domain knowledge and the RDF data. A list of currently available TogoStanza used in the TogoGenome database (**Supplemental Table 3.1** and **Appendix**) can be found at <http://togostanza.org/> where users can try out their functionality by changing the arguments on the fly. Additionally, NanoStanza is another form of TogoStanza that summarizes information at a glance in an icon-sized module (**Figure 3.4**). The metadata of each TogoStanza is written in the JSON-LD format and is used to automatically summarize and categorize each TogoStanza in the showcase page.

To date, more than 250 TogoStanza have been developed, including those developed for databases other than the TogoGenome database (**Table 3.1**). The TogoStanza framework is well suited to web application development, especially for Semantic Web data in various life sciences and biomedical domains. In BioJS (Corpas *et al.*, 2014), which is a similar web application framework, that is not specialized for the Semantic Web, 195 components are provided. Among these components, only one module (nextprot-cli) seems to use SPARQL.

Using TogoStanza, components that are common to several databases in the life sciences

and biomedical domains are successfully modularized, leading to a reduction in the development costs and making the resulting database extensible for new functionalities.

Table 3.1 List of TogoStanza providers.

Database	Domain	Number of TogoStanza	URL
TogoGenome	Genome	59	http://togogenome.org/stanza/
MicrobeDB.jp	Genome	113	http://microbedb.jp/stanza/
CyanoBase	Genome	6	http://genome.microbedb.jp/stanza/
MBGD	Ortholog	19	http://mbgd.genome.ad.jp/stanza/
GlyTouCan	Glycomics	16	https://bitbucket.org/glycosw/glytoucan-stanza/ https://github.com/glytoucan/glytoucan-js-stanza/
jPOST	Proteomics	15	http://tools.jpostdb.org/ts/stanza/
TogoVar	Variation	26	https://togovar.biosciencedbc.jp/stanza

Embed TogoStanza JavaScript version

```
<!doctype html>
<html>
  <body>
    <link rel="import" href="//togostanza.org/dist/example/" />
    <togostanza-example ec="3.1.-.-" />
  </body>
</html>
```

Embed TogoStanza Ruby version

```
<!doctype html>
<html>
  <head>
    <script src="//code.jquery.com/jquery-3.3.1.js" />
    <script src="http://togostanza.org/stanza/assets/stanza.js" />
  </head>
  <body>
    <div data-stanza="http://togostanza.org/stanza/protein_names"
          data-stanza-tax-id="9606" data-stanza-gene-id="ALDH2" />
  </body>
</html>
```

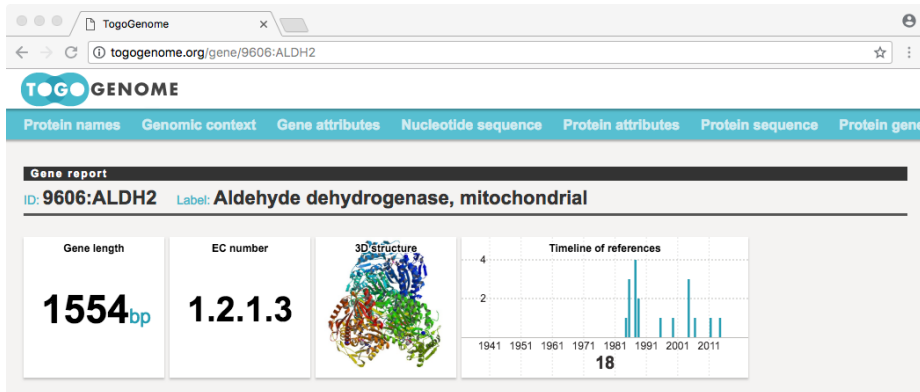
The screenshot shows the TogoStanza web interface for the gene ALDH2. The page title is "TogoStanza GENOME". The main content area displays a "Gene report" for "ID: 9606:ALDH2" with the label "Aldehyde dehydrogenase, mitochondrial". Key statistics include a gene length of 1554 bp, an EC number of 1.2.1.3, and a 3D protein structure. A "Timeline of references" chart shows 18 references from 1941 to 2011. Below the statistics is a "Protein names" table with the following data:

Protein names	Recommended Name	Aldehyde dehydrogenase, mitochondrial
	EC number	1.2.1.3
	Alternative Name(s)	<ul style="list-style-type: none">• ALDH class 2• ALDH-E2• ALDHI
Gene names		<ul style="list-style-type: none">• ALDH2
	Synonyms	<ul style="list-style-type: none">• ALDM
Organism	Homo sapiens	
Taxonomic identifier	9606	
Taxonomic lineage	<ul style="list-style-type: none">• cellular organisms• Eukaryota• Opisthokonta	

Figure 3.3 Embedding TogoStanza into a web page.

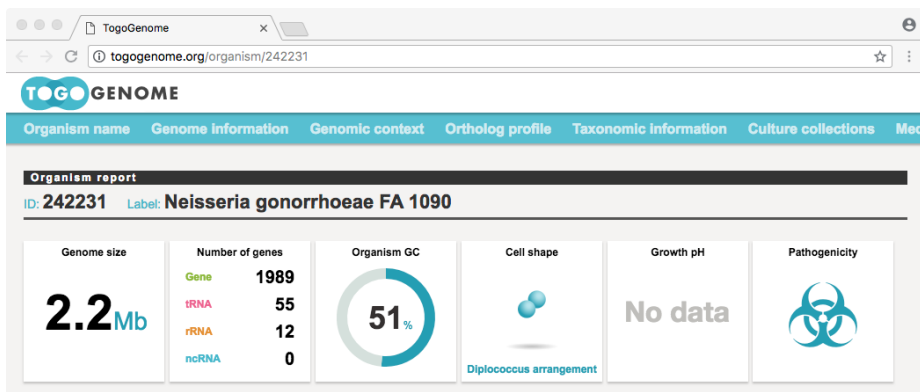
NanoStanza for gene

Gene length, Enzyme number, 3D structure, Publications per year



NanoStanza for organism

Genome size, Number of genes, GC content, Cell shape, Growth pH, Pathogenicity



NanoStanza for environment

Habitat, Number of inhabitants, Distribution of growth temperature and pH

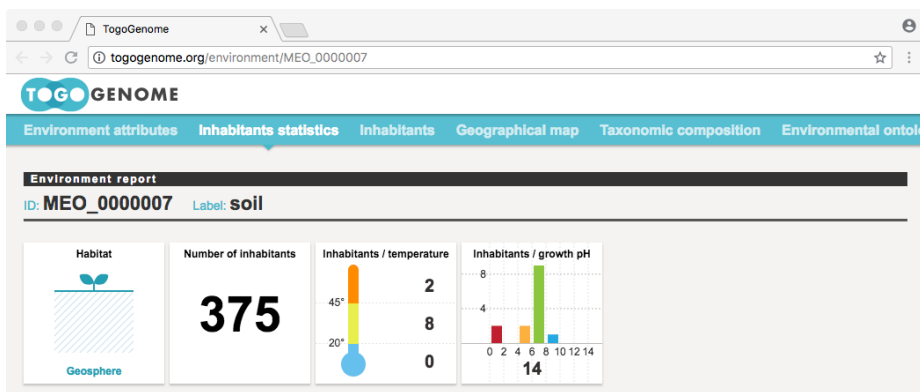
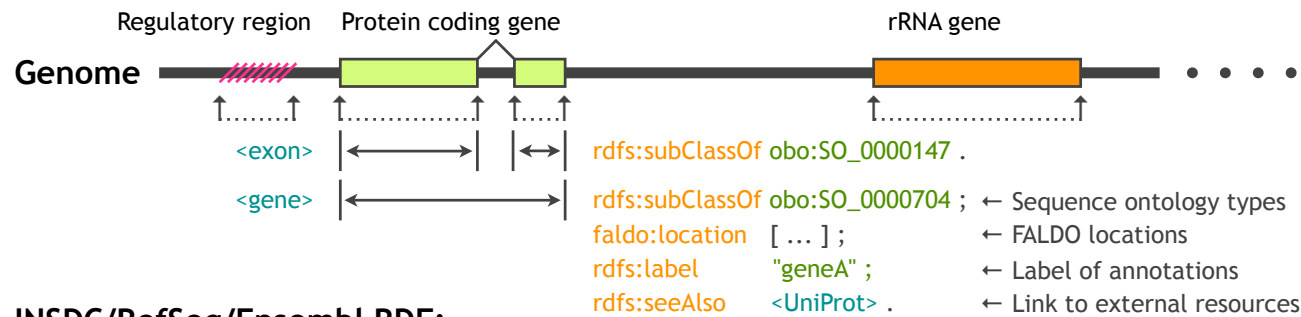


Figure 3.4 NanoStanza in gene, organism, and environment report pages.

3.3 Methods

3.3.1 Integration of genome annotations

Any annotations related to genome regions, such as gene structures, regulatory regions, mutations, and modifications, can be located using the genomic coordinate system. This information can be integrated by uniquely identifying the reference sequence, specifying the beginning and terminating positions of the region to which the annotation is attached, and designating the type of the annotation. However, if the ontologies and the RDF data model to describe these feature locations are not standardized, a query for one genome database cannot be interoperable with another even if the genome annotations are provided in RDF. For this reason, during the BioHackathon 2013 (<http://2013.biohackathon.org/>) (Katayama *et al.*, 2014) and the RDF summit (<https://github.com/dbcls/rdfsummit>) coding events, I collaboratively developed the Feature Annotation Location Description Ontology (FALDO) (Bolleman *et al.*, 2016) together with the UniProt, Ensembl, INSDC (DDBJ), and TogoGenome groups (**Figure 3.5**). The JBrowse genome browser version 1.10.0 (Buels *et al.*, 2016) was also developed to implement a SPARQL query for acquiring and visualizing the annotations expressed using FALDO. Traditionally, several standards, such as GFF (<http://gmod.org/wiki/GFF3>) and TrackHubs (Raney *et al.*, 2014), have been developed to attach annotations to the genome coordinates. The same can be achieved in RDF using FALDO along with some other major ontologies such as the Sequence Ontology (SO) (Mungall *et al.*, 2011) and the SemanticScience Integrated Ontology (SIO) (Dumontier *et al.*, 2014). Therefore, it is possible to construct a genome browser that can be of practical application while ensuring compatibility of the annotation information among the genome datasets represented in RDF.



INSDC/RefSeq/Ensembl RDF:






<gene>	rdf:type so:so_part_of	insdc:Gene ; <chromosome> .	
<mRNA>	rdf:type sio:is-transcribed-from sio:has-ordered-part	insdc:Messenger_RNA ; <gene> ; <p1>, <p2>,	    
<p1>	sio:has-value sio:refers-to	"1"^^xsd:integer ; <exon1> .	
<p2>	sio:has-value sio:refers-to	"2"^^xsd:integer ; <exon2> .	
<exon1>	rdf:type faldo:location	insdc:Exon ; <region1> .	
<region1>	rdf:type faldo:begin faldo:end	faldo:Region ; <position1> ; <position2> .	
<position1>	rdf:type faldo:position faldo:reference	faldo:ExactPosition, faldo:ForwardStrandPosition ; 12345 ; <chromosome> .	

Figure 3.5 Standardization of the genome annotation coordinate system by the FALDO ontology.

3.3.2 TogoGenome datasets

On the basis of the above standardization, my collaborators and I jointly developed the following RDF datasets and ontologies to integrate the public resources in TogoGenome (**Figure 3.6**).

Complete genomes: We selected the "reference genome" and "representative genome" entries from the NCBI assembly report and further extracted RefSeq and Taxonomy identifiers.

Genome annotations: We retrieved the NCBI RefSeq entries, including entire chromosome sequences, via the TogoWS service (Katayama, Nakao, *et al.*, 2010a). Further, each entry was converted to RDF using an in-house converter, which is based on the BioRuby library (Goto *et al.*, 2010a) and represents the feature locations using FALDO. To semantically describe the types of annotations, we developed and incorporated the INSDC annotated sequence ontology (**Table 3.2**) along with the taxonomy ontology described below. This converter is now publicly available (**Table 3.2**); it is used to publish the RDF version of the INSDC entries from DDBJ (Mashima *et al.*, 2017) and is hosted at the NBDC RDF portal (**Table 3.2**).

Genome sequences: We extracted the genome sequences from the RefSeq entries and further indexed them for the JBrowse genome browser and the GGGenome sequence search service.

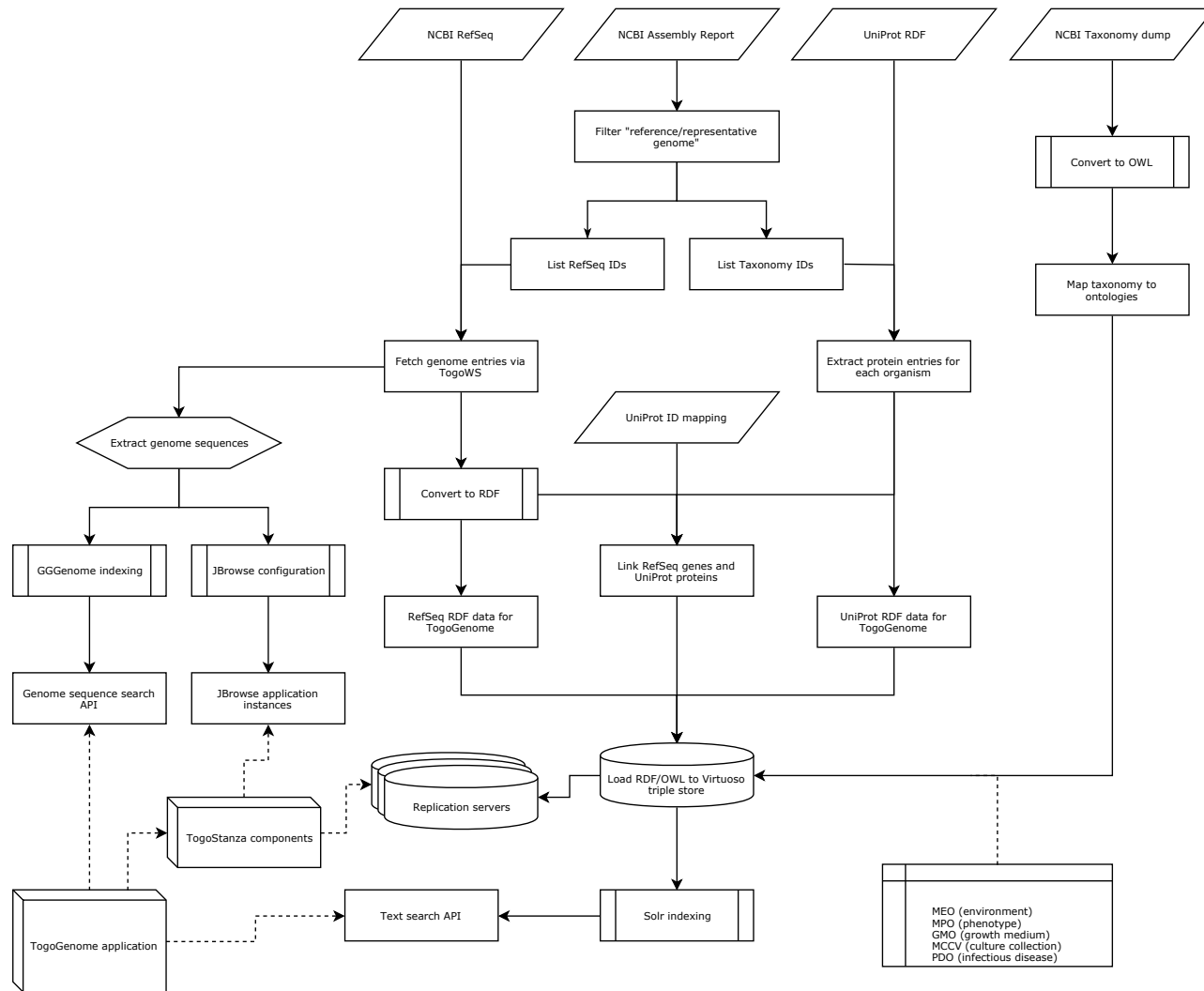


Figure 3.6 Procedure of data integration in TogoGenome.

Taxonomy: We obtained a taxonomy dump from NCBI that contained all the species that were recorded in the INSDC sequence archive and their taxonomic hierarchies. Further, using an in-house converter (**Table 3.2**), we converted the dump to an OWL ontology file. The resulting ontology is publicly available (**Table 3.2**) and is used in the INSDC (DDBJ) RDF export.

Protein information: We obtained the UniProt RDF files and extracted protein entries belonging to species with complete genomes. Meanwhile, genes in RefSeq were mapped with UniProt proteins using UniProt's idmapping file. Technically, it is possible to directly use the UniProt SPARQL endpoint, however, the performance of SPARQL federated queries was not satisfactory for our purpose and we only needed a subset of the entire UniProt database. Therefore, we imported a portion of the relevant UniProt data into TogoGenome.

In-house ontologies: We developed the Microbial Phenotype Ontology (MPO) for microbial phenotypes, Metagenome and Microbes Environmental Ontology (MEO) for habitat environments, Microbial Culture Collection Vocabulary (MCCV) for culture collections, Growth Medium Ontology (GMO) for growth media and Pathogenic Disease Ontology (PDO) for infectious diseases (**Table 3.2**). These ontologies were mapped onto the taxonomy ontology.

Other ontologies: We used FALDO for annotation coordinates, SO and INSDC annotated sequence ontology for the types of annotated regions, and Gene Ontology (GO) for gene functions along with common ontologies such as SIO, Dublin Core terms (DC), and Simple Knowledge Organization System (SKOS).

As of June 2018, TogoGenome has integrated 7,065 complete genome sequences of 2,196 organisms (212 eukaryotes), which include 10,843,971 genes (4,070,521 eukaryotic genes), along with their corresponding UniProt protein annotations. In total, approximately 6.3 billion triples of RDF data are stored and updated upon every RefSeq/UniProt release. The RDF database system, which is a triple store, that is

currently being used is the Virtuoso open source version 7 (<http://vos.openlinksw.com/>), and it is scalable at least up to tens of billion triples in our experience. To improve the response of the SPARQL endpoint, the stored RDF data file in a single loading instance is copied to three backend Virtuoso instances (16GB of each of the RAMs are allocated) for load balancing at the Nginx HTTP server layer. With this configuration, I can eliminate service downtime during the update procedure by sequentially updating and restarting these backend servers. This SPARQL endpoint is publicly available at <http://togogenome.org/sparql> for accepting customized queries from users.

Table 3.2 Availability of in-house converters and ontologies.

RDF data, ontologies and converters	URL
INSDC RDF hosted at the NBDC RDF Portal	https://integbio.jp/rdf/
INSDC annotated sequence ontology	http://ddbj.nig.ac.jp/ontologies/nucleotide/
INSDC/RefSeq record to RDF converter	https://github.com/dbcls/rdfsummit/tree/master/insdc2ttl/
INSDC taxonomy ontology	http://ddbj.nig.ac.jp/ontologies/taxonomy/
NCBI taxonomy to INSDC taxonomy converter	https://github.com/dbcls/rdfsummit/tree/master/taxdump2owl/
Microbial Phenotype Ontology (MPO)	https://bioportal.bioontology.org/ontologies/MPO
Metagenome and Microbes Environmental Ontology (MEO)	https://bioportal.bioontology.org/ontologies/MEO
Microbial Culture Collection Vocabulary (MCCV)	https://bioportal.bioontology.org/ontologies/MCCV
Growth Medium Ontology (GMO)	https://bioportal.bioontology.org/ontologies/GMO
Pathogenic Disease Ontology (PDO)	https://bioportal.bioontology.org/ontologies/PDO

3.3.3 Development of TogoGenome

The TogoGenome application itself has been built using Ruby on Rails (<https://rubyonrails.org/>). Functions such as faceted search, comparative genomics, and keyword and sequence searches, are implemented in this application layer. For the faceted search, we use several ontologies in combination such as (1) GO annotations imported from UniProt RDF for gene features, (2) NCBI taxonomy that has been converted to OWL and released at DDBJ for organisms, (3) MPO that has been developed for phenotypes, and (4) MEO that has been made for the habitat. Candidate ontology terms will be suggested while keywords are being typed, and the user can traverse the hierarchy of ontologies to adjust the granularity of classification. To improve the performance of the faceted search, I calculated in advance the correspondences between higher-level concepts in ontologies and genes that fall under the categories and stored the inferred relations at the time of updating the data. Additionally, the combinations of the selected facets are stored in a user's cookie, and the query results are cached as much as possible to improve response time.

While SPARQL queries are suitable for semantic searches of interconnected objects in the RDF datasets, the efficiency of character string and regular expression searches is inefficient in most of the triple stores. In TogoGenome, I introduced the Apache Solr full-text search system for indexing character strings such as names, descriptions, and other text-based annotations of genes and organisms. However, identifying the page on which a searched term is displayed without tracing the connections between triples and pages was still difficult. To resolve this issue, we selected the targeted fields for the text searches in each TogoStanza and further indexed the strings and corresponding stanzas in pairs. For example, in the case of a gene report page, currently seven stanzas contain literal annotations of a gene, each representing different aspects of the same gene. I therefore created an index that contains a TogoGenome gene URI and a literal string for each TogoStanza (**Figure 3.7**). In this figure, the gene URI indicated by @id and a literal string containing the information about IDs and annotations were used in indexing. This indexing procedure is iterated over all genes of each organism stored in TogoGenome.

```

{
  "@id": "http://togogenome.org/gene/9606:ALDH2",
  "gene_id": "9606:ALDH2",
  "uniprot_id": [ "P05091" ],
  "names": [ "Caution", "Polymorphism", "Similarity", "Subcellular Location", "Subunit" ],
  "messages": [
    "Belongs to the aldehyde dehydrogenase family.",
    "Genetic variation in ALDH2 is responsible for individual differences in responses to drinking alcohol [MIM:610251] ... ",
    "Homotetramer.",
    "Mitochondrion matrix",
    "No experimental confirmation available."
  ]
}

```

Figure 3.7 Correspondence of TogoGenome URI and literal strings for each TogoStanza to be indexed for text search in Apache Solr.

(This shows an example of human ALDH2 protein annotation in the “Protein general annotation” stanza).

Similarly, searching for genomic regions that have a specific sequence with SPARQL is not efficient. Therefore, I used the GGGenome system (<https://GGGenome.dbcls.jp/>) API to obtain the corresponding chromosome and its position. Using the specified sequence ID and location, genomic annotations around the region can be obtained by a SPARQL query using the FALDO ontology.

3.3.4 Development of TogoStanza

Due to historical reasons, there are two branches of the TogoStanza development framework. TogoStanza was originally developed as a Ruby application but was later implemented using JavaScript. Both of these branches are able to generate template files for SPARQL and HTML along with the files for metadata and supporting data.

The Ruby version of the TogoStanza framework was released as a RubyGems' package (<https://rubygems.org/>). Therefore, users can install it via Ruby's standard 'gem' command and further generate the TogoStanza template files using the installed 'togostanza' command. After customizing the templates and developing the query and visualization logic, the resulting TogoStanza can be deployed at the TogoStanza server and embedded into any web page as an IFRAME. Because the IFRAME encapsulates its content, other elements on a web page, even on a classical web browser, are not affected. However, due to the strict isolation of IFRAME, it is difficult to make a TogoStanza interact with other TogoStanza even if both contain components that are embedded on the same page. Further, this version requires the TogoStanza process to keep running on the server while the SPARQL queries that are implemented inside the TogoStanza are executed on the server side. Therefore, a heavy load may be created while exhibiting exceeding accesses. This problem can be resolved in the JavaScript version of TogoStanza.

The JavaScript version of the TogoStanza module relies only on standard web technologies, such as HTML, CSS, JavaScript, AJAX, and SPARQL, and generates Web Components as a static HTML file. This eliminates the dependency on the server side where the SPARQL queries are made via an AJAX call directly from the user's web browser to the public SPARQL endpoint. The results are rendered by the client browser.

Using “Web Components” technology, which encapsulates Document Object Model (DOM) as a shadow DOM, multiple TogoStanza can be embedded in a single DOM of a web page so that it is possible to implement components that react to an event that has been issued by another component upon a user's interaction. The current drawback is that the state of the browser's support, even while using a modern web browser, is not optimized for Web Components. Therefore, it will take a while for the transition from the Ruby version to the JavaScript version. Therefore, I provide a special Ruby version TogoStanza that wraps the JavaScript version as a temporal countermeasure.

3.4 Discussion

By introducing a modularized architecture for displaying SPARQL results as TogoStanza, I and our collaborators were able to reduce the costs of mutually constructing new genome-related databases in TogoGenome, MicrobeDB.jp, MBDG (Uchiyama *et al.*, 2015), and CyanoBase. This exchange of distributed resources could not be achieved by existing monolithic genome database systems. The idea of providing reusable application components based on standard web technology is a natural extension of the concept of the Semantic Web. In the Semantic Web, RDF data stored in distributed SPARQL endpoints are transparently accessible through the standard HTTP/HTTPS protocol unlike the data buried in intranet database systems. Therefore, it is possible to use distributed heterogeneous data on a reciprocal basis. Additionally, RDF is scalable for the integration of heterogeneous data types without being bound by the database schema.

Traditionally, most genome databases are built on top of high-performance database engines such as a relational database (RDB) or key-value stores. I was unsure about the performance of emerging triple stores for RDF. Further, the original version of TogoGenome had been implemented using other triple stores or prior versions of Virtuoso and had not scaled enough in the beginning. However, the Virtuoso open source version 7, released in 2013, exhibited sufficiently high practical performance for our genome database by making tens of real-time SPARQL queries at once against billions of triples.

I have successfully demonstrated an RDF back-ended system with real-time SPARQL

queries that can be used for a large-scale genome database. Meanwhile, I observed that triple stores were not efficient for text searches. However, this is not necessarily a defect of the Semantic Web system. Even while using relational databases or other NoSQL databases, it is the norm to prepare a text search engine to perform keyword searches and an external application, such as BLAST, to perform sequence searches.

Traditional databases required users to parse a database entry to extract information, forcing them to develop custom scripts with programming language-dependent open source libraries, such as BioPerl (Stajich *et al.*, 2002), Biopython (Cock *et al.*, 2009), BioJava (Prlić *et al.*, 2012), and BioRuby (Goto *et al.*, 2010a), before performing real bioinformatics analyses. For databases that do not publish flat file dumps, a web interface that can retrieve the summarized information is often provided. However, the flexibility and granularity of information that can be obtained by users are usually restricted by the capability of the provided APIs. In RDF, all the information is already parsed and semantically annotated. In the case of SPARQL, especially with the ontologies and adaptable conditions, it is relatively straightforward to obtain any aggregated information by filtering data.

3.5 Conclusions

I introduced a modularized architecture in the TogoGenome database that allowed database developers to reuse the typical annotations of genes and organisms in other organism-specific or metagenome databases as embeddable TogoStanza components (**Supplemental Table 3.1** and **Appendix**).

Because all the RDF data, the SPARQL endpoint, and TogoStanza components that are used in the TogoGenome application are publicly available, developers who intend to build another genome database will benefit from the usage of these resources to reduce the costs of application development and data management costs.

Chapter 4

Discussions and conclusions

In order to realize integrated utilization of life science databases, I conducted research for improving standardization and interoperability of database access methods and database contents. Since realization of such standardization cannot be achieved by a single institute, it was necessary to collaborate with an international community to develop systems and semantic datasets.

Originally, I have developed the BioRuby library supporting a number of database formats and bioinformatics applications. This library has been used to develop a client program to conduct data analysis and create reproducible workflows. However, it turned out that a workflow which utilizes Web services faced difficulties in connecting the output of one service into the input of the next service because of data type incompatibility. Also, APIs in existing Web services vary in its form of calling APIs and accepting data formats, thus requiring users to consult the documentation of each service and to develop data conversion programs. This situation gave me the idea to standardize the APIs of these services and to improve the interoperability of Web services by creating a new Web service, TogoWS, which fills this gap of incompatibility.

The mission of Database Center for Life Science (DBCLS) is to promote integrated use of life science databases. However, because of the exponential growth in volume of these databases, it is becoming hard to maintain a centralized database at a single center. Instead, it is more efficient and sustainable to virtually integrate distributed databases. For this purpose, Web services is one applicable technology, and my TogoWS development described above can contribute to realize this integration. During the course of development, I also added support for RDF conversion in TogoWS which exposes database contents as Linked Data.

By providing an RDF version of data, information retrieval and analysis based on the meaning of data, which was difficult with conventional databases, could be realized. Therefore, the next mission of the DBCLS became the advancement of standardized database contents in the life sciences and biomedical domains using Semantic Web technologies. With this shift, the possibility of new data usage which could not be realized by conventional technologies was greatly increased. In TogoGenome, I collaboratively integrated heterogeneous datasets in the genomics domain, such as organisms, genes, proteins, phenotypes, and environments, as RDF data and ontologies. As a result, users could benefit from a faceted search system using multiple ontologies in combination to semantically extract information of interest. This kind of information retrieval could not be achieved by conventional database systems thus demonstrating one of the advantages of Semantic Web technologies.

In addition to create and maintain billions of triples in an RDF database, genomics data often contain tremendous volumes of raw data such as SAM/BAM sequence read alignments, epigenomic data and variant calling data. I realized that semantic integration of data is most suitable for information such as facts and annotations including gene structures, locations and protein functions as in the case of a gene report. In contrast, raw data supporting this information, which is usually displayed in a genome browser, do not necessarily need to be represented as RDF, because they are usually stored in an efficient binary format for improving read/write performance.

Another improvement in TogoGenome was to introduce a modularized structure. TogoGenome is composed of a combination of multiple TogoStanzas, each of which displays an information block resulting from their respective SPARQL queries. A new TogoStanza is first initialized with templates including a code snippet with scaffold SPARQL and HTML files. The developer can override the SPARQL queries and the logic of data transformation for visualization. The resultant data is rendered in HTML and sent back to the client Web browser. A drawback of this implementation design is that useful SPARQL queries are buried in the existing TogoStanza instances and are not easily

reusable. Therefore, I started to externalize these SPARQL queries as REST APIs using SPARQList (<https://github.com/dbcls/sparqlist>), which is a new Web service I created at DBCLS for improving the reusability of SPARQL-based REST APIs. In SPARQList, a developer can describe an API itself in Markdown format with documentation, embed SPARQL queries, and logic to transform SPARQL results into convenient JSON data. The resulting API is instantly deployed and can be executed from a Web interface and from any Web client applications especially through an AJAX call as in TogoStanza. Therefore, by exposing well-developed SPARQL queries in TogoStanza as SPARQList, users can easily reuse complex queries for their analysis, and advanced users can test and modify these queries for similar purposes.

For future research, I initiated the development of a human genome variation database, TogoVar (<https://togovar.biosciencedbc.jp/>), in which the human subset of TogoGenome is reused. Then my colleagues and I in the TogoVar project added information about genomic variations and allele frequencies in the Japanese population. Thanks to Semantic Web technologies, it is relatively easy to extend data models in TogoGenome to integrate new type of datasets in the TogoVar database. Therefore, I also started the Med2RDF project (<http://med2rdf.org/>) to develop RDF datasets of biomedical resources with colleagues, which integrates knowledge of genetic diseases, sequence and structural variants, cancer genomes, protein and drug interactions, and clinical significance. All the RDF data we developed are being accumulated and hosted in the NBDC RDF Portal (Kawashima *et al.*, 2018). These resources will be essential for future data science research, and it is anticipated that new methods will be developed by introducing advanced analytical techniques such as machine learning.

References

- Benson,D.A. *et al.* (2017) GenBank. *Nucleic Acids Res.*, **45**, D37–D42.
- Bolleman,J.T. *et al.* (2016) FALDO: a semantic standard for describing the location of nucleotide and protein feature annotation. *J. Biomed. Semantics*, **7**, 39.
- Buels,R. *et al.* (2016) JBrowse: a dynamic web platform for genome visualization and analysis. *Genome Biol.*, **17**, 66.
- Cochrane,G. *et al.* (2016) The International Nucleotide Sequence Database Collaboration. *Nucleic Acids Res.*, **44**, D48-50.
- Cock,P.J.A. *et al.* (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, **25**, 1422–3.
- Corpas,M. *et al.* (2014) BioJS: an open source standard for biological visualisation - its status in 2014. *F1000Research*, **3**, 55.
- Dehal,P. *et al.* (2002) The draft genome of *Ciona intestinalis*: insights into chordate and vertebrate origins. *Science*, **298**, 2157–67.
- Dumontier,M. *et al.* (2014) The SemanticScience Integrated Ontology (SIO) for biomedical research and knowledge discovery. *J. Biomed. Semantics*, **5**, 14.
- Fielding,R.T. (2000) Architectural Styles and the Design of Network-based Software Architectures.
- Fujisawa,T. *et al.* (2017) CyanoBase: a large-scale update on its 20th anniversary. *Nucleic Acids Res.*, **45**, D551–D554.
- Goto,N. *et al.* (2010a) BioRuby: bioinformatics software for the Ruby programming language. *Bioinformatics*, **26**, 2617–9.
- Goto,N. *et al.* (2010b) BioRuby: Bioinformatics software for the Ruby programming language. *Bioinformatics*, **26**, 2617–2619.
- Holland,R.C.G. *et al.* (2008) BioJava: an open-source framework for bioinformatics. *Bioinformatics*, **24**, 2096–7.
- Hull,D. *et al.* (2006) Taverna: a tool for building and running workflows of services. *Nucleic Acids Res.*, **34**, W729-32.
- Kalderimis,A. *et al.* (2014) InterMine: extensive web services for modern biology. *Nucleic Acids Res.*, **42**, W468-72.
- Kanehisa,M. *et al.* (2010) KEGG for representation and analysis of molecular networks involving diseases and drugs. *Nucleic Acids Res.*, **38**, D355-60.
- Katayama,T. *et al.* (2014) BioHackathon series in 2011 and 2012: penetration of

- ontology and linked data in life science domains. *J. Biomed. Semantics*, **5**, 5.
- Katayama, T. *et al.* (2011) The 2nd DBCLS BioHackathon: interoperable bioinformatics Web services for integrated applications. *J. Biomed. Semantics*, **2**, 4.
- Katayama, T. *et al.* (2013) The 3rd DBCLS BioHackathon: improving life science data integration with semantic Web technologies. *J. Biomed. Semantics*, **4**, 6.
- Katayama, T., Arakawa, K., *et al.* (2010) The DBCLS BioHackathon: standardization and interoperability for bioinformatics web services and workflows. The DBCLS BioHackathon Consortium*. *J. Biomed. Semantics*, **1**, 8.
- Katayama, T. *et al.* (2019) TogoGenome/TogoStanza: modularized Semantic Web genome database. *Database (Oxford)*, **2019**, 1–11.
- Katayama, T., Nakao, M., *et al.* (2010a) TogoWS: integrated SOAP and REST APIs for interoperable bioinformatics Web services. *Nucleic Acids Res.*, **38**, W706-11.
- Katayama, T., Nakao, M., *et al.* (2010b) TogoWS: integrated SOAP and REST APIs for interoperable bioinformatics Web services. *Nucleic Acids Res.*, **2008**, 1–6.
- Kawashima, S. *et al.* (2018) NBDC RDF portal: a comprehensive repository for semantic data in life sciences. *Database (Oxford)*, **2018**, 1–11.
- Kwon, Y. *et al.* (2009) Web API for biology with a workflow navigation system. *Nucleic Acids Res.*, **37**, W11-6.
- Labarga, A. *et al.* (2007) Web services at the European bioinformatics institute. *Nucleic Acids Res.*, **35**, W6-11.
- Mashima, J. *et al.* (2017) DNA Data Bank of Japan. *Nucleic Acids Res.*, **45**, D25–D31.
- Mishima, H. *et al.* (2012) The Ruby UCSC API: accessing the UCSC genome database using Ruby. *BMC Bioinformatics*, **13**, 240.
- Miyazaki, S. *et al.* (2004) DDBJ in the stream of various biological data. *Nucleic Acids Res.*, **32**, D31-4.
- Mungall, C.J. *et al.* (2011) Evolution of the Sequence Ontology terms and relationships. *J. Biomed. Inform.*, **44**, 87–93.
- O'Connor, B.D. *et al.* (2008) GMODWeb: a web framework for the Generic Model Organism Database. *Genome Biol.*, **9**, R102.
- O'Leary, N.A. *et al.* (2016) Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Res.*, **44**, D733-45.
- Pillai, S. *et al.* (2005) SOAP-based services provided by the European Bioinformatics Institute. *Nucleic Acids Res.*, **33**, W25-8.
- Prlić, A. *et al.* (2012) BioJava: an open-source framework for bioinformatics in 2012. *Bioinformatics*, **28**, 2693–5.

- Raney,B.J. *et al.* (2014) Track data hubs enable visualization of user-defined genome-wide annotations on the UCSC Genome Browser. *Bioinformatics*, **30**, 1003–5.
- Sayers,E.W. *et al.* (2009) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.*, **37**, D5-15.
- Silvester,N. *et al.* (2018) The European Nucleotide Archive in 2017. *Nucleic Acids Res.*, **46**, D36–D40.
- Stajich,J.E. *et al.* (2002) The Bioperl toolkit: Perl modules for the life sciences. *Genome Res.*, **12**, 1611–8.
- Stajich,J.E. and Lapp,H. (2006) Open source tools and toolkits for bioinformatics: significance, and where are we? *Brief. Bioinform.*, **7**, 287–96.
- Standley,D.M. *et al.* (2008) Protein structure databases with new web services for structural biology and biomedical research. *Brief. Bioinform.*, **9**, 276–85.
- Stockinger,H. *et al.* (2008) Experience using web services for biological sequence analysis. *Brief. Bioinform.*, **9**, 493–505.
- Sugawara,H. *et al.* (2008) DDBJ with new system and face. *Nucleic Acids Res.*, **36**, D22-4.
- Sugawara,H. and Miyazaki,S. (2003) Biological SOAP servers and web services provided by the public sequence data bank. *Nucleic Acids Res.*, **31**, 3836–9.
- The UniProt Consortium (2017) UniProt: the universal protein knowledgebase. *Nucleic Acids Res.*, **45**, D158–D169.
- Uchiyama,I. *et al.* (2015) MGD update 2015: microbial genome database for flexible ortholog analysis utilizing a diverse set of genomic data. *Nucleic Acids Res.*, **43**, D270-6.
- UniProt Consortium (2008) The universal protein resource (UniProt). *Nucleic Acids Res.*, **36**, D190-5.
- Vandervalk,B.P. *et al.* (2009) Moby and Moby 2: creatures of the deep (web). *Brief. Bioinform.*, **10**, 114–28.
- Wilkinson,M.D. and Links,M. (2002) BioMOBY: an open source biological web services proposal. *Brief. Bioinform.*, **3**, 331–41.
- Zerbino,D.R. *et al.* (2018) Ensembl 2018. *Nucleic Acids Res.*, **46**, D754–D761.

Supplemental Figures and Tables

Supplemental Table 2.1 List of available databases for keyword search. The first column represents canonical database names and the second is for aliases if defined.

pdj-pdb	pdb
kegg-compound	compound
kegg-drug	drug
kegg-enzyme	enzyme
kegg-genes	genes
kegg-glycan	glycan
kegg-orthology	orthology
kegg-reaction	reaction
kegg-module	module
kegg-pathway	pathway
ncbi-pubmed	pubmed
ncbi-protein	protein
ncbi-nuccore	nuccore
ncbi-nucleotide	nucleotide
ncbi-nucgss	nucgss
ncbi-nucest	nucest
ncbi-structure	
ncbi-genome	
ncbi-assembly	
ncbi-gcassembley	
ncbi-genomeprj	
ncbi-bioproject	
ncbi-biosample	
ncbi-biosystems	
ncbi-blastdbinfo	
ncbi-books	
ncbi-cdd	
ncbi-clone	
ncbi-gap	

ncbi-gapplus
ncbi-dbvar
ncbi-epigenomics
ncbi-gene gene
ncbi-gds
ncbi-geoprofiles
ncbi-homologene homologene
ncbi-journals
ncbi-medgen
ncbi-mesh mesh
ncbi-ncbisearch
ncbi-nlmcatalog
ncbi-omia
ncbi-omim omim
ncbi-pmc
ncbi-popset
ncbi-probe
ncbi-proteinclusters
ncbi-pcassay
ncbi-pccompound
ncbi-pcsubstance
ncbi-pubmedhealth
ncbi-seqannot
ncbi-snp snp
ncbi-sra
ncbi-taxonomy
ncbi-toolkit
ncbi-toolkitall
ncbi-toolkitbook
ncbi-unigene
ncbi-unists
ncbi-gencoll
ebi-arrayexpress-repository
ebi-atlas-experiments
ebi-atlas-genes
ebi-biomodels

ebi-chebi
ebi-chembl-activity
ebi-chembl-assay
ebi-chembl-target
ebi-dgva
ebi-efo
ebi-ega
ebi-emblnew_con
ebi-emblnew_standard
ebi-emblrelease_con
ebi-emblrelease_standard
ebi-ensemblGenomes_gene
ebi-ensembl_gene
ebi-epo
ebi-genome_assembly
ebi-go
ebi-gpcrdb
ebi-hgnc
ebi-intact-experiments
ebi-intact-interactions
ebi-intact-interactors
ebi-intenz
ebi-interpro
ebi-jpo
ebi-kipo
ebi-lrg
ebi-medline
ebi-merops_clan
ebi-merops_family
ebi-merops_id
ebi-nrn11
ebi-nrn12
ebi-nrpl1
ebi-nrpl2
ebi-omim
ebi-patentFamilies

ebi-patentdb
ebi-pdbe
ebi-pdbechem
ebi-pride
ebi-project
ebi-reactome
ebi-rhea
ebi-sbo
ebi-sra-analysis
ebi-sra-experiment
ebi-sra-run
ebi-sra-sample
ebi-sra-study
ebi-sra-submission
ebi-taxonomy
ebi-uniparc uniparc
ebi-uniprot uniprot
ebi-uniref100 uniref100
ebi-uniref50 uniref50
ebi-uniref90 uniref90
ebi-uspto
ebi-wgs_masters

Supplemental Table 2.2 List of available databases for entry retrieval. The first column represents canonical database names and the second is for aliases.

ncbi-nuccore	nuccore
ncbi-nucest	nucest
ncbi-nucgss	nucgss
ncbi-nucleotide	nucleotide
ncbi-protein	protein
ncbi-gene	gene
ncbi-homologene	homologene
ncbi-snp	snp
ncbi-mesh	mesh
ncbi-pubmed	pubmed
ebi-ena	ena
ebi-uniprot	uniprot
ebi-uniparc	uniparc
ebi-uniref100	uniref100
ebi-uniref90	uniref90
ebi-uniref50	uniref50
ddbj-ddbj	ddbj
ddbj-dad	dad
pdbj-pdb	pdb
kegg-compound	compound
kegg-drug	drug
kegg-enzyme	enzyme
kegg-genes	genes
kegg-glycan	glycan
kegg-orthology	orthology
kegg-reaction	reaction
kegg-module	module
kegg-pathway	pathway

Supplemental Table 2.3 List of available pairs of a source data type and a converted format. The first part before a period represents an acceptable source data type and the latter part shows a convertible format.

genbank.fasta
genbank.ena
genbank.gff
genbank.ntriples
genbank.n3
genbank.rdfxml
genbank.ttl
ena.fasta
ena.genbank
ena.ntriples
ena.n3
ena.rdfxml
ena.ttl
ddbj.ntriples
ddbj.n3
ddbj.rdfxml
ddbj.ttl
uniprot.fasta
uniprot.gff
blast.gff
blasttable.gff
blastxml.gff
megablast.gff
fasta.gff
psl.gff
sim4.gff
hmmer.gff
hmmer3tbl.rdfxml
exonerate.gff
wise.gff
rdfxml.ttl
ttl.rdfxml

csv.rdfxml

csv.ttl

gff.rdfxml

gff.ttl

gvf.rdfxml

gvf.ttl

Supplemental Table 3.1 List of TogoStanzas developed for the TogoGenome.

Genes and proteins

gene_attributes: basic information of a gene
nucleotide_sequence: nucleotide sequence of a gene
protein_attributes: basic information of a protein
protein_names: canonical and alternative names of a protein
protein_general_annotation: functional annotations of a protein
protein_orthologs: links to orthologous proteins
protein_cross_references: cross references to other protein resources
protein_ontologies: keywords and gene ontology annotations
protein_references: links to literature
protein_sequence: information and amino acid sequence of a protein
protein_sequence_annotation: domain and functional sites of a protein

Genomes

genome_information: list of chromosomal sequences
genome_cross_references: cross references to other genome resources
genome_jbrowse: genome browser
genome_plot: scatter plot of a distribution of genomic properties

Organisms

organism_names: list of organism names and synonyms
organism_phenotype: phenotype descriptions of a organism
organism_cross_references: cross references to organism related resources
organism_gene_list: list of genes of a organism
lineage_information: taxonomic lineage of an organism
organism_habitat: list of habitats of a organism
organism_pathogen_information: organism related pathogenic diseases
organism_culture_collections: summary of culture collections of a organism
organism_medium_information: medium information of a organism
taxonomy_ortholog_profile: taxonomic profile of an ortholog group

Environments

environment_attributes: description of an environment
environment_environmental_ontology: hierarchical view of an environment
environment_geographical_map: geographic locations of an environment
environment_inhabitants: samples and cultures taken from an environment
environment_inhabitants_statistics: number of samples and cultures
environment_taxonomic_composition: breakdown of organisms of an environment

Phenotypes

microbial_phenotype_information: list of organisms of a phenotype
microbial_phenotype_cell_shape: description of a shape of a microbial
microbial_phenotype_environment_composition: environments and phenotype
microbial_phenotype_genus_composition: genus and phenotype

Medium

gmo_applied_spices: applied species of a medium
gmo_approximation: relevance among medium
gmo_genus: medium based organism summary
medium_components: components of a medium

NanoStanza

gene_length_nano
gene_wikidata_nano
protein_ec_number_nano
protein3d_structure_nano
protein_references_timeline_nano
organism_gene_number_nano
organism_genome_size_nano
organism_gc_nano
organism_microbial_cell_shape_nano
organism_related_disease_nano
organism_wikidata_nano

organism_ph_nano

environment_inhabitants_statistics_nano

environment_organism_distribution_on_ph_nano

environment_organism_distribution_on_temperature_nano

environment_top_level_symbolic_image_nano

MetaStanza

js_stanza_wrapper

Appendix

TogoWS API specification

In this section, generic functionalities and advanced features of the TogoWS APIs are described.

TogoWS REST API conventions

TogoWS REST API currently supports following functionalities.

Entry [http://togows.org/entry/database/entry_id\[,entry_id2,...\]\[/?field\]\[.format\]](http://togows.org/entry/database/entry_id[,entry_id2,...][/?field][.format])

Search [http://togows.org/search/database/query+string\[/offset,limit\]\[.format\]](http://togows.org/search/database/query+string[/offset,limit][.format])

Convert http://togows.org/convert/data_source.format

External API [http://togows.org/api/service/database/table/column=value\[/offset,limit\]\[.format\]](http://togows.org/api/service/database/table/column=value[/offset,limit][.format])

Entry

Entry retrieval REST API can be used to obtain database entries, extract a field content and convert the data format.

- Synopsis
 - [http://togows.org/entry/database/entry_id\[,entry_id2,...\]\[/?field\]\[.format\]](http://togows.org/entry/database/entry_id[,entry_id2,...][/?field][.format])
- Multiple entries
 - Multiple entries can be retrieved at once by concatenating identifiers with ',' (100 entries at maximum)
- Options
 - **List of available databases:** <http://togows.org/entry/database> (some have alias for short in the second column)
 - **List of available fields:** <http://togows.org/entry/database?fields>
 - **List of available formats:** <http://togows.org/entry/database?formats>
 - **Splice of a nucleotide sequence:** <http://togows.org/entry/database/seq/location>
- Errors

- **400 Bad Request** (HTTP error): requested URI or the **database** was invalid
- **404 Not Found** (HTTP error): requested entry was not found

Search

Database search REST API can be used to obtain a list of entry identifiers or a number of results by a keyword search.

- Synopsis
 - [http://togows.org/search/database/query+string\[/offset,limit\]\[.format\]](http://togows.org/search/database/query+string[/offset,limit][.format])
- Query string
 - Format of the "query string" is just a simple text (spaces can be replaced with '+' or '%20')
- Options
 - **List of available databases:** <http://togows.org/search/> (some have alias for short in the second column)
 - **List of available formats:** <http://togows.org/search/database?format>
 - **Limit the number of results:** http://togows.org/search/database/query+string/offset_limit
 - **Count the number of results:** <http://togows.org/search/database/query+string/count>
- Errors
 - **400 Bad Request** (HTTP error): requested URI or the **database** was invalid
 - **404 Not Found** (HTTP error): requested entry had no results

Convert

Data format conversion REST API can be used to convert file formats.

- Synopsis
 - http://togows.org/convert/data_source.format
- Protocol
 - Use the HTTP POST protocol to upload your data as a text
- Options
 - **List of available converters:** <http://togows.org/convert/> (description of these formats can be found [here](#) and [here](#))
- Errors
 - **400 Bad Request** (HTTP error): requested URI was invalid
 - **404 Not Found** (HTTP error): requested **data_source.format** is not supported

TogoWS external API

External API is introduced to provide REST APIs for accessing non-Web service based external data sources, such as University of California, Santa Cruz (UCSC) databases.

UCSC API

UCSC API internally uses the [Ruby UCSC API](#) library which directly accesses to the public MySQL database provided by the UCSC. Although UCSC uses 0-based inter-base coordinates, TogoWS accepts biological 1-based positions (which can contain commas but not mandatory) and are automatically converted when accessing the UCSC database.

- Synopsis
 - `http://togows.org/api/ucsc/database/table/column[!]=value[;column2[!]=value2;...]/[offset,limit][.format]`
- Options
 - **List of available databases:** `http://togows.org/api/ucsc/.json`
 - **List of available tables:** `http://togows.org/api/ucsc/database/.json`
 - **List of available columns:** `http://togows.org/api/ucsc/database/table/.json` (example data will be shown in the second column)
 - **Obtain a limited number of results:** `http://togows.org/api/ucsc/database/table/offset,limit.json`
- Keyword search
 - **Find rows having a keyword in a given column:** `http://togows.org/api/ucsc/database/table/column=value/[offset,limit].json`
 - **Find rows not having a keyword in a given column:** `http://togows.org/api/ucsc/database/table/column!=value/[offset,limit].json`
 - **Filtering by multiple conditions:** `http://togows.org/api/ucsc/database/table/column[!]=value[;column2[!]=value2;...]/[offset,limit].json`
- Range search
 - **Find rows within a range:** `http://togows.org/api/ucsc/database/table/chromosome:from-to.json` (default to inclusive)
 - **Include rows straddle over a boundary:** `http://togows.org/api/ucsc/database/table/inclusive/chromosome:from-to.json`

- **Exclude rows straddle over a boundary:** [http://togows.org/api/ucsc/database/table/exclusive/chromosome:from-to\[.json\]](http://togows.org/api/ucsc/database/table/exclusive/chromosome:from-to[.json])
- Gene table (refGene, knownGene, ensGene, wgEncode etc.)
 - **CDS positions:** [http://togows.org/api/ucsc/database/refGene/name\[2\]=value/cds\[.json\]](http://togows.org/api/ucsc/database/refGene/name[2]=value/cds[.json])
 - **Exon positions:** [http://togows.org/api/ucsc/database/refGene/name\[2\]=value/exon\[.json\]](http://togows.org/api/ucsc/database/refGene/name[2]=value/exon[.json])
 - **Intron positions:** [http://togows.org/api/ucsc/database/refGene/name\[2\]=value/intron\[.json\]](http://togows.org/api/ucsc/database/refGene/name[2]=value/intron[.json])
- bigWig data
 - **Corresponding file name:** <http://togows.org/api/ucsc/database/bigWig>
 - **bigWigInfo output:** <http://togows.org/api/ucsc/database/bigWig/info>
 - **bigWigSummary output:** <http://togows.org/api/ucsc/database/bigWig/chromosome:from-to/divisor> (default to mean)
 - **Mean:** <http://togows.org/api/ucsc/database/bigWig/chromosome:from-to/divisor/mean>
 - **Minimum:** <http://togows.org/api/ucsc/database/bigWig/chromosome:from-to/divisor/min>
 - **Maximum:** <http://togows.org/api/ucsc/database/bigWig/chromosome:from-to/divisor/max>
 - **Coverage:** <http://togows.org/api/ucsc/database/bigWig/chromosome:from-to/divisor/coverage>
 - **Standard deviation:** <http://togows.org/api/ucsc/database/bigWig/chromosome:from-to/divisor/std>
- bigBed data
 - **Corresponding file name:** <http://togows.org/api/ucsc/database/bigBed>
 - **bigBedInfo output:** <http://togows.org/api/ucsc/database/bigBed/info>
 - **bigBedSummary output:** <http://togows.org/api/ucsc/database/bigBed/chromosome:from-to/divisor> (default to coverage)
 - **Mean:** <http://togows.org/api/ucsc/database/bigBed/chromosome:from-to/divisor/mean>
 - **Minimum:** <http://togows.org/api/ucsc/database/bigBed/chromosome:from-to/divisor/min>
 - **Maximum:** <http://togows.org/api/ucsc/database/bigBed/chromosome:from-to/divisor/max>
 - **Coverage:** <http://togows.org/api/ucsc/database/bigBed/chromosome:from-to/divisor/coverage>
- DNA sequence (2bit file)
 - **Forward strand:** [http://togows.org/api/ucsc/database/chromosome:from-to\[.fasta\]](http://togows.org/api/ucsc/database/chromosome:from-to[.fasta])
 - **Reverse strand:** [http://togows.org/api/ucsc/database/chromosome:to-from\[.fasta\]](http://togows.org/api/ucsc/database/chromosome:to-from[.fasta])

TogoWS API examples

In this section, representative examples of the TogoWS APIs are shown.

TogoWS entry retrieval API examples

Retrieve a PubMed entry 20472643.

```
% curl http://togows.org/entry/pubmed/20472643
PMID- 20472643
OWN - NLM
STAT- MEDLINE
DCOM- 20100927
LR - 20141203
IS - 1362-4962 (Electronic)
IS - 0305-1048 (Linking)
VI - 38
IP - Web Server issue
DP - 2010 Jul
TI - TogoWS: integrated SOAP and REST APIs for interoperable bioinformatics Web
    services.
PG - W706-11
:
(60 lines)
```

Retrieve a PubMed entry 20472643 and extract authors in a JSON format.

```
% curl http://togows.org/entry/pubmed/20472643/authors.json
[
  [
    "Katayama, T.",
    "Nakao, M.",
    "Takagi, T."
  ]
]
```

Retrieve a UniProt entry BRCA2_HUMAN.

```
% curl http://togows.org/entry/uniprot/BRCA2_HUMAN
ID BRCA2_HUMAN Reviewed; 3418 AA.
AC P51587; O00183; O15008; Q13879; Q5TBJ7;
DT 01-OCT-1996, integrated into UniProtKB/Swiss-Prot.
DT 11-NOV-2015, sequence version 3.
DT 12-SEP-2018, entry version 208.
DE RecName: Full=Breast cancer type 2 susceptibility protein;
DE AltName: Full=Fanconi anemia group D1 protein;
GN Name=BRCA2; Synonyms=FACD, FANCD1;
OS Homo sapiens (Human).
:
(1750 lines)
```

Retrieve UniProt entries ACT_YEAST and ACT_SCHPO in a FASTA format.

```
% curl http://togows.org/entry/uniprot/ACT_YEAST,ACT_SCHPO.fasta
>sp|P60010|ACT_YEAST Actin OS=Saccharomyces cerevisiae (strain ATCC 204508 / S288c) OX=559292
GN=ACT1 PE=1 SV=1
MDSEVAALVIDNGSGMCKAGFAGDDAPRAVFPISVGRPRHQGIMVGMGQKDSYVGDEAQS
KRGILTLRYPYIEHGIVTNWDDMEKIWHHTFYNELRVAPPEHPVLLTEAPMNPKNREKMT
QIMFETFNVPAFYVSIQAVLSLYSSGRTTGIVLDSGDGVTHVVPYIYAGFSLPHAILRIDL
AGRDLTDYLMKILSERGYSFSTTAEREIVRDIKEKLCYVALDFEQEMQTAQAQSSSIEKSY
ELPDGQVITIGNERFRAPEALFHPSVLGLESAGIDQTTYNSIMKCDVDVRKELYGNIVMS
GGTTMFPGIAERMQKEITALAPSSMKVKIAPPKYSVWIGGSILASLTFQQMWISKQ
EYDESGPSIVHHKCF
>sp|P10989|ACT_SCHPO Actin OS=Schizosaccharomyces pombe (strain 972 / ATCC 24843) OX=284812
GN=act1 PE=1 SV=1
MEEETIAALVIDNGSGMCKAGFAGDDAPRAVFPISVGRPRHHGIMVGMGQKDSYVGDEAQS
KRGILTLKYPYIEHGIVNNWDDMEKIWHHTFYNELRVAPPEHPCLLTEAPLNPKSNREKMT
QIIFETFNAPAFYVAIQAVLSLYASGRTTGIVLDSGDGVTHVTPYIEGYALPHAIMRLDL
AGRDLTDYLMKILMERYTFSTTAEREIVRDIKEKLCYVALDFEQELQTAQAQSSSLEKSY
ELPDGQVITIGNERFRAPEALFQPSALGLENAGIHEATYNSIMKCDVDIRKDLGNVMS
GGTTMYPGIADRMQKEIQALAPSSMKVKIVAPPKYSVWIGGSILASLSTFQQMWISKQ
EYDESGPGIVYRKCF
```

Retrieve a RefSeq entry NC_001138 (yeast chromosome 6) in a GFF format.

```
% curl http://togows.org/entry/nucleotide/NC_001138.gff
##gff-version 3
NC_001138 Genbank region 1 270161 . . .
ID=NC_001138;Note=Saccharomyces%20cerevisiae%20S288C%20chromosome%20VI%2C%20complete%20
sequence.
NC_001138 Genbank region 1 270161 . + .
ID=Saccharomyces%20cerevisiae%20S288C;db_xref=taxon%3A559292;chromosome=VI;strain=S288C
;mol_type=genomic%20DNA
NC_001138 Genbank telomere 1 5530 . - .
ID=TEL06L%3B%20Telomeric%20region%20on%20the%20left%20arm%20of%20Chromosome%20VI%3B%20c
omposed%20of%20an%20X%20element%20core%20sequence%2C%20X%20element%20combinatorial%20repeats%2C%
20a%20stretch%20of%20telomeric%20repeats%2C%20and%20a%20short%20Y%27%20element;db_xref=SGD%3AS00
0028882
NC_001138 Genbank gene 53 535 . + .
ID=YFL068W;db_xref=GeneID%3A850476
NC_001138 Genbank mRNA 53 535 . + .
Parent=YFL068W;ID=YFL068W.t01;db_xref=GeneID%3A850476;transcript_id=NM_001179899.1;prod
uct=hypothetical%20protein
NC_001138 Genbank CDS 53 535 . + .
Parent=YFL068W.t01;protein_id=NP_116587.1;note=hypothetical%20protein%3B%20SWAT-
GFP%20and%20mCherry%20fusion%20proteins%20localize%20to%20the%20cytosol;codon_start=1;db_xref=Ge
neID%3A850476,SGD%3AS000001826;translation=MMPAKLQLDVLRLTLQSSARHGTQTLKNSNFLERFHKDRIVFCLPFFPALFLVP
VQKVLQHLCLRFTQVAPYFIIQLFDLPSRHAENLAPLLASCRIQYTNCFSSSSNGQVPSIISLYLRVDLSPFYAKKFQIPYRVPMIWLDVQVFFV
FLVISQHSLSHS;product=hypothetical%20protein
:
(4996 lines)
```

Extract a sub-sequence that of the ACT1 gene which has two exons at 53260..54377 and 54687..54696 on a reverse complement strand of the RefSeq entry NC_001138 (yeast chromosome 6).

```
% curl 'http://togows.org/entry/nucleotide/NC_001138/seq/complement(join(53260..54377,54687..
54696))'
```

```
atggattctgaggttgctgctttggttattgataacggttctgggatgtgtaaagccggttttgcggtgacgacgctcctctgctgtcttccca
tctatcgtcggttagaccaagacaccaaggtatcatggtcggtatgggtcaaaaagactcctacgttggatgaagctcaatccaagagaggtatc
ttgactttacgttaccattgaacacggtattgtcaccactgggacgatatggaaaagatctggcatcataccttctacaacgaattgagagtt
gccccagaagaacaccctgttcttttgactgaagctcaatgaaccctaaatcaaacagagaaaagatgactcaaattatgtttgaaactttcaac
gttccagccttctacgtttccatccaagccgtttgtccttgactcttcggtagaactactggatattgttttgattccggtgatgggttact
cacgtcgttccaatttacgtggtttctctctacctcacgccattttgagaatcgatttggccggtagagatttgactgactacttgatgaagatc
ttgagtgaacgtggttactcttctccaccactgctgaaagagaaattgtccgtgacatcaaggaaaaactatgttacgtcgccttgacttcgaa
caagaaatgcaaacctgctcaatcttctcaattgaaaaatcctacgaacttcagatgggtcaagtcatcactattggtaacgaaagattcaga
gccccagaagctttgttccatccttctgttttgggtttggaatctgccggtattgaccaaactacttacaactccatcatgaagtgtgatgcgat
gtccgtaaggaattatacggtaacatcgttatgtccggtggtaccaccatgttccaggatttgcgaaagaatgcaaaaggaaatcaccgctttg
gctccatcttccatgaaggtcaagatcattgtcctccagaagaaagtactccgtctggattggtggttctatcttggcttctttgactacctc
caacaaatgtggatctcaaaacaagaatacgcgaaagtgggtccatctatcgttcaccacaagtgtttctaa
```

Show a list of available formats for the NCBI Nucleotide database.

```
% curl 'http://togows.org/entry/nucleotide?formats'
gb
xml
ttl
fasta
gff
json
```

Show a list of available fields for the NCBI Nucleotide database.

```
% curl 'http://togows.org/entry/nucleotide?fields'
entry_id
length
strand
moltype
linearity
division
date
definition
accession
accessions
```

version
versions
acc_version
gi
keywords
organism
common_name
taxonomy
comment
seq
references
features

TogoWS search API examples

Search the UniProt database using the phrase “lung cancer”.

```
% curl http://togows.org/search/uniprot/lung+cancer
KKLC1_HUMAN
DLEC1_HUMAN
KKLC1_MACFA
Q7Z5Q7_HUMAN
A0A0A8K8N9_HUMAN
A0A0A8K9B1_HUMAN
A0A0A8K8F0_HUMAN
A0A0A8K8C0_HUMAN
A0A0A8K9A6_HUMAN
ALDOA_HUMAN
HOP_HUMAN
MED19_HUMAN
RBM6_HUMAN
S22AI_HUMAN
S38A9_HUMAN
:
```

Search the UniProt database using the phrase “lung cancer” and retrieve the first five entry IDs.

```
% curl http://togows.org/search/uniprot/lung+cancer/1,5
KKLC1_HUMAN
DLEC1_HUMAN
KKLC1_MACFA
Q7Z5Q7_HUMAN
A0A0A8K8N9_HUMAN
```

Search the next five entry IDs and return the results in a JSON format.

```
% curl http://togows.org/search/uniprot/lung+cancer/6,5.json
["A0A0A8K9B1_HUMAN", "A0A0A8K8F0_HUMAN", "A0A0A8K8C0_HUMAN", "A0A0A8K9A6_HUMAN", "ALDOA_HUMAN"]
```

Search the next five entry IDs and return the results in an HTML format.

```
% curl http://togows.org/search/uniprot/lung+cancer/11,5.json
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
</head>
<body>
<div><a href="http://togows.org/entry/uniprot/HOP_HUMAN" target="_blank">HOP_HUMAN</a></div>
<div><a href="http://togows.org/entry/uniprot/MED19_HUMAN" target="_blank">MED19_HUMAN</a></div>
<div><a href="http://togows.org/entry/uniprot/RBM6_HUMAN" target="_blank">RBM6_HUMAN</a></div>
<div><a href="http://togows.org/entry/uniprot/S22AI_HUMAN" target="_blank">S22AI_HUMAN</a></div>
<div><a href="http://togows.org/entry/uniprot/S38A9_HUMAN" target="_blank">S38A9_HUMAN</a></div>
</body>
</html>
```

Count the number of search results in the UniProt database using the phrase “lung cancer”.

```
% curl http://togows.org/search/uniprot/lung+cancer/count
449
```

TogoWS convert API examples

First, prepare a GenBank entry J00231 to be converted.

```
% wget http://togows.org/entry/nucleotide/J00231
```

```
% ls
```

```
J00231
```

Convert the obtained file into a GFF format via HTTP POST.

```
% wget http://togows.org/convert/genbank.gff --post-file=J00231 -O J00231.gff
```

```
% ls
```

```
J00231    J00231.gff
```

```
% head J00231
```

```
LOCUS      HUMIGHAF              1089 bp   mRNA   linear   PRI 09-NOV-1994
DEFINITION Human Ig gamma3 heavy chain disease OMM protein mRNA.
ACCESSION  J00231
VERSION    J00231.1
KEYWORDS   C-region; V-region; gamma heavy chain disease protein; gamma3 heavy
           chain disease protein; heavy chain disease; hinge exon;
           immunoglobulin gamma-chain; immunoglobulin heavy chain; secreted
           immunoglobulin.
SOURCE     Homo sapiens (human)
           ORGANISM Homo sapiens
```

```
% cat J00231.gff
```

```
##gff-version 3
```

```
J00231    Genbank   region   1         1089     .         .         .
           ID=J00231;Note=Human%20Ig%20gamma3%20heavy%20chain%20disease%20OMM%20protein%20mRNA.
J00231    Genbank   region   1         1089     .         +         .
           ID=Homo%20sapiens;map=14q32.33;mol_type=mRNA;db_xref=taxon%3A9606
J00231    Genbank   gene     1         1089     .         +         .         ID=IGHG3
```


J00231 Genbank mRNA 1 1089 . + .
Parent=IGHG3;ID=IGHG3.t01;product=gamma3%20mRNA

J00231 Genbank CDS 23 964 . + .
Parent=IGHG3.t01;protein_id=AAA52805.1;note=OMM%20protein%20%28Ig%20gamma3%29%20heavy%20chain;db_xref=GDB%3AG00-119-339;codon_start=1;translation=MKXLWFFLLLVAAAPRWVLSQVHLQESGPGLGKPELKTPLGDTTHTCPRCPEPKSCDTPPPCPRCP
EPKSCDTPPPCPRCPEPKSCDTPPPCPCXPAPELLGGPSVFLFPPKPKDTLMISRTPEVTCVVVDVSHEDPXVQFKWYVDGVEVHNAKTKLREEQY
NSTFRVSVLTVLHQDWLNGKEYKCKVSNKALPAPIEKISKAKGQXXXXXXXXXXXXEEMTKNQVSLTCLVKGFYPSDIAVEWESNGQPENNYN
TTPMMLSDSGSFLLYSKLTVDKSRWQQGNIFSCSVMHEALHNRYTQKLSLSLSPGK

J00231 Genbank sig_peptide 26 79 . + .
ID=IGHG3;note=OMM%20protein%20signal%20peptide

J00231 Genbank mat_peptide 80 961 . + .
ID=IGHG3;product=OMM%20protein%20mature%20peptide

>J00231
cctggacctcctgtgcaagaacatgaaacantgtggttcttcttctcctggtggcagc
tcccagatgggtcctgtcccaggtgcacctgcaggagtggggccaggactggggaagcc
tccagagctcaaaacccacttggtgacacaactcacacatgccacgggtgccagagcc
caaatctgtgacacacctccccgtgcccaggtgcccagagccaaatcttgtgacac
acctccccatgccacgggtgccagagccaaatcttgtgacacacctccccgtgccc
nngtgcccagcacctgaactcttgggaggaccgtcagtccttcttcccccaaaacc
caaggatacccttatgattcccggacccctgaggtcacgtgcgtggtggtggactgag
ccagaaagaccnngtccagttcaagtggtacgtggacggcgtggaggtgataatgc
caagaaaagctcgggaggagcagtacaacagcacgttccgtgtggtcagcgtcctcac
cgtcctgcaccaggactggctgaacggcaaggagtacaagtgcaaggtctcaacaaagc
cctcccagccccatcgagaaaaccatctccaaagccaaaggacagccnnnnnnnnnn
nnnnnnnnnnnnnnnnnnnnngaggatgaccaagaaccaagtcagcctgacctg
cctggtcaaaggcttctaccccagcgacatcgccgtggagtgggagagcaatgggcagcc
ggagaacaactacaacaccacgctcccatgctggactccgacggctccttcttctcta
cagcaagctcaccgtggacaagagcaggtggcagcaggggaacatcttctcatgctccgt
gatgcatgaggctctgcacaaccgctacacgcagaagagcctctccctgtctccgggtaa
atgagtgccatggccggaagccccgctccccgggctctcggggtcgcgcgaggatgct
tggcagctacccgtgtacatacttcccaggcaccagcatggaaataaagcaccagcg
ctgccctgg

TogoWS external API examples

Show a list of available UCSC genome databases.

```
% curl http://togows.org/api/ucsc/
```

```
ailMe11  
anoCar2  
anoGam1  
apiMe12  
aplCa11  
bosTau4  
braFlo1  
caeJap1  
caePb3  
caeRem3  
calJac3  
canFam2  
cavPor3  
cb3  
ce6  
ci2  
danRer10  
danRer11  
danRer7  
dm3  
dp3  
droAna2  
droEre1  
droGri1  
droMoj2  
droPer1  
droSec1  
droSim1  
droVir2  
droYak2
```

equCab2
felCat4
fr2
galGal3
gasAcu1
go
hg18
hg19
hg38
hgFixed
loxAfr3
mm10
mm9
monDom5
ornAna1
oryCun2
oryLat2
oviAri1
panTro3
petMar1
ponAbe2
priPac1
proteome
rheMac2
rn4
rn5
sacCer2
strPur2
susScr2
taeGut1
tetNig2
uniProt
visiGene
xenTro2

Show a list of available tables in the hg38 database.

```
% curl http://togows.org/api/ucsc/hg38/  
affyGnf1h  
affyU133  
affyU95  
all_est  
all_mrna  
all_sts_primer  
all_sts_seq  
altLocations  
altSeqLiftOverPs1  
altSeqLiftOverPs1P11  
:  
(903 lines)
```

Show columns of the refGene table of the hg38 database with values of the first record.

```
% curl http://togows.org/api/ucsc/hg38/refGene/  
bin      1815  
name     NR_110164  
chrom    chr2  
strand   +  
txStart  161244738  
txEnd    161249050  
cdsStart 161249050  
cdsEnd   161249050  
exonCount 2  
exonStarts      161244738,161246874,  
exonEnds 161244895,161249050,  
score     0  
name2     LINC01806  
cdsStartStat      unk  
cdsEndStat        unk  
exonFrames        -1,-1,
```

Retrieve the first five records from the refGene table of the hg38 database.

```
% curl http://togows.org/api/ucsc/hg38/refGene/1,5
```

bin	name	chrom	strand	txStart	txEnd	cdsStart	cdsEnd	exonCount	exonStarts	exonEnds	score	name2	cdsStartStat	cdsEndStat	exonFrames
1815	NR_110164	chr2	+	161244738	161249050	161249050	161249050	2		161244738,161246874,			161244895,161249050,	0	LINC01806 unk
	unk		-1,-1,												
27	NR_110250	chr2	-	156020534	156254931	156254931	156254931	4		156020534,156022671,156024465,156254777,			156021899,156022817,156024607,156254931,	0	LINC01876 unk
															unk -
															1,-1,-1,-1,
585	NR_128720	chr16	-	17051	17119	17119	17119	1	17051,	17119,	0	MIR6859-4 unk	unk	unk	-1,
637	NR_128718	chr21	+	6859170	6859256	6859256	6859256	1	6859170,	6859256,	0	MIR8069-2 unk	unk	unk	-1,
689	NR_128718	chr21	+	13724188	13724274	13724274	13724274	1	13724188,	13724274,	0	MIR8069-2 unk	unk	unk	-1,

Retrieve the next five records in a JSON format from the refGene table of the hg38 database.

```
% curl http://togows.org/api/ucsc/hg38/refGene/6,5.json
```

```
[{"bin":585,"name":"NR_128718","chrom":"chrUn_GL00213v1","strand":"-","txStart":25282,"txEnd":25368,"cdsStart":25368,"cdsEnd":25368,"exonCount":1,"exonStarts":25282,"exonEnds":25368,"score":0,"name2":"MIR8069-2","cdsStartStat":"unk","cdsEndStat":"unk","exonFrames":"-1"}, {"bin":647,"name":"NR_128717","chrom":"chr21","strand":"+","txStart":8205314,"txEnd":8205406,"cdsStart":8205406,"cdsEnd":8205406,"exonCount":1,"exonStarts":8205314,"exonEnds":8205406,"score":0,"name2":"MIR6724-4","cdsStartStat":"unk","cdsEndStat":"unk","exonFrames":"-1"}, {"bin":647,"name":"NR_128717","chrom":"chr21","strand":"+","txStart":8249504,"txEnd":8249596,"cdsStart":8249596,"cdsEnd":8249596,"exonCount":1,"exonStarts":8249504,"exonEnds":8249596,"score":0,"name2":"MIR6724-4","cdsStartStat":"unk","cdsEndStat":"unk","exonFrames":"-1"}, {"bin":648,"name":"NR_128717","chrom":"chr21","strand":"+","txStart":8388361,"txEnd":8388453,"cdsStart":8388453,"cdsEnd":8388453,"exonCount":1,"exonStarts":8388361,"exonEnds":8388453,"score":0,"name2":"MIR6724-4","cdsStartStat":"unk","cdsEndStat":"unk","exonFrames":"-1"}, {"bin":1599,"name":"NR_049862","chrom":"chr9","strand":"+","txStart":132945706,"txEnd":1329
```

```
45771,"cdsStart":132945771,"cdsEnd":132945771,"exonCount":1,"exonStarts": "132945706, ", "exonEnds"
:"132945771, ", "score":0,"name2": "MIR548AW", "cdsStartStat": "unk", "cdsEndStat": "unk", "exonFrames":
"-1, "}]
```

Retrieve records having a gene name UVSSA in the name2 column of the refGene table of the hg38 database.

```
% curl http://togows.org/api/ucsc/hg38/refGene/name2=UVSSA
```

bin	name	chrom	strand	txStart	txEnd	cdsStart	cdsEnd	exonCount	exonStarts	exonEnds	score	name2	cdsStartStat	cdsEndStat	exonFrames
595	NM_020894	chr4	+	1347315	1388049	1348091	1385961	14	1347315,1348089,1349523,1351714,1353029,1354734,1355116,1366319,1375363,1376033,1380046,1380879,1383765,1385867,1347760,1348189,1349854,1351835,1353413,1354847,1355245,1366431,1375508,1376168,1380230,1380988,1383940,1388049,	0	UVSSA	cmp1	cmp1	-1,0,2,0,1,1,0,0,1,2,2,0,1,2,	
595	NM_001317934	chr4	+	1347265	1388049	1348091	1385961	14	1347265,1348089,1349523,1351714,1353029,1354734,1355116,1366319,1375363,1376033,1380046,1380879,1383765,1385867,1347609,1348189,1349854,1351835,1353413,1354847,1355245,1366431,1375508,1376168,1380230,1380988,1383940,1388049,	0	UVSSA	cmp1	cmp1	-1,0,2,0,1,1,0,0,1,2,2,0,1,2,	
595	NM_001317935	chr4	+	1347555	1388049	1348091	1385961	14	1347555,1348089,1349523,1351714,1353029,1354734,1355116,1366319,1375363,1376033,1380046,1380879,1383765,1385867,1347887,1348189,1349854,1351835,1353413,1354847,1355245,1366431,1375508,1376168,1380230,1380988,1383940,1388049,	0	UVSSA	cmp1	cmp1	-1,0,2,0,1,1,0,0,1,2,2,0,1,2,	

Retrieve the first five records on the chromosome 13 of the hg18 database.

```
% curl http://togows.org/api/ucsc/hg38/refGene/chrom=chr13/1,5
```

bin	name	chrom	strand	txStart	txEnd	cdsStart	cdsEnd	exonCount	exonStarts	exonEnds	score	name2	cdsStartStat	cdsEndStat	exonFrames
0	NM_203487	chr13	-	66302833	67230336	66304654	67228440	5	66302833,66631209,66903503,67225404,67229779,66305028,66631411,66903605,67228575,67230336,	0	PCDH9	cmp1	cmp1	1,0,0,0,-1,	

```

0      NM_020403 chr13      -      66302833 67230336 66304654 67228440 4
      66302833,66631209,67225404,67229779, 66305028,66631411,67228575,67230336, 0
      PCDH9      cpl      cpl      1,0,0,-1,
0      NM_001318373      chr13      -      66302833 67230336 66304654 67228440 4
      66302833,66631209,67225404,67229779, 66305028,66631285,67228575,67230336, 0
      PCDH9      cpl      cpl      1,0,0,-1,
0      NM_001318372      chr13      -      66302833 67230336 66304654 67228440 5
      66302833,66631209,66903503,67225404,67229779,
      66305028,66631285,66903605,67228575,67230336, 0      PCDH9      cpl      cpl
      1,0,0,0,-1,
1      NM_199138 chr13      -      25161678 25172167 25169603 25171619 3
      25161678,25170838,25171912, 25170481,25171719,25172167, 0      AMER2      cpl
      cpl      1,0,-1,

```

Retrieve the first five records having an allele frequency count other than 0 in the dbSNP version 138 for which reference allele is an adenine on the reference human chromosome 22 of the hg19 genome build.

```
% curl 'http://togows.org/api/ucsc/hg19/snp138/chrom=chr22;refUCSC=A;alleleFreqCount!=0/1,5'
```

```

bin      chrom      chromStart      chromEnd name      score      strand      refNCBI      refUCSC
observed molType      class      valid      avHet      avHetSE      func      locType      weight
exceptions      submitterCount      submitters      alleleFreqCount      alleles
alleleNs      alleleFreqs      bitfields
707      chr22      16050374 16050375 rs2844882 0      +      A      A      A/G
genomic      single      by-cluster,by-2hit-2allele 0.0      0.0      0      exact      3
0      6      BCM-HGSC-SUB,BCM HGSC_JDW,ENSEMBL,SC_JCM,SSAHASNP,WI_SSAHASNP,
A,      6.000000,1.000000,
707      chr22      16050739 16050740 rs111307625      0      +      A      A      -
/A      genomic      deletion      unknown      0.5      0.0      0      exact      3      0      1
BUSHMAN, 2      -,A,      1.000000,1.000000, 0.500000,0.500000,
707      chr22      16051208 16051209 rs7292503 0      +      A      A      A/G
genomic      single      by-cluster,by-2hit-2allele,by-hapmap 0.5      0.0      0
exact      1      0      4      BCM_SSAHASNP,COMPLETE_GENOMICS,CSHL-
HAPMAP,WI_SSAHASNP,2      A,G,      1.000000,1.000000, 0.500000,0.500000,

```

```

707 chr22 16051391 16051392 rs77125914 0 - A A
C/T genomic single unknown 0.5 0.0 0 exact 3 0
1 ENSEMBL, 2 C,T, 1.000000,1.000000, 0.500000,0.500000,
707 chr22 16051452 16051453 rs143503259 0 + A A
A/C genomic single by-cluster,by-1000genomes 0.135347 0.222159 0
exact 1 0 2 1000GENOMES,SSMP, 2 A,C,
2019.000000,159.000000, 0.926997,0.073003,

```

Retrieve the dbSNP version 138 records overlapping with a region from 20,000 to 21,000bp on the chromosome 1 of the hg19 genome build.

```
% curl http://togows.org/api/ucsc/hg19/snp138/chr1:20,000-21,000
```

```

bin chrom chromStart chromEnd name score strand refNCBI refUCSC
observed molType class valid avHet avHetSE func locType weight
exceptions submitterCount submitters alleleFreqCount alleles
alleleNs alleleFreqs bitfields
585 chr1 20036 20037 rs12354133 0 + A A
A/C genomic single unknown 0.0 0.0 0 exact 3 0
2 SC_SNP,SSAHASNP, 0
585 chr1 20043 20044 rs75790700 0 + C C
C/T genomic single unknown 0.5 0.0 0 exact 3 0
1 ENSEMBL, 2 C,T, 1.000000,1.000000, 0.500000,0.500000,
585 chr1 20127 20128 rs806718 0 - G G C/T
genomic single unknown 0.0 0.0 0 exact 3 0 3
KWOK,SC_JCM,TSC-CSHL, 0
585 chr1 20127 20128 rs75128330 0 + G G
A/G genomic single unknown 0.5 0.0 0 exact 3 0
1 ENSEMBL, 2 A,G, 1.000000,1.000000, 0.500000,0.500000,
585 chr1 20127 20128 rs111753557 0 - G G
C/T genomic single unknown 0.5 0.0 0 exact 3 0
1 BUSHMAN, 2 C,T, 1.000000,1.000000, 0.500000,0.500000,

```

```
:
```

```
(116 lines)
```

Retrieve the refGene records overlapping with a region from 1,350,000 to 1,400,000bp on the chromosome 4 of the hg38 genome build.


```
% curl http://togows.org/api/ucsc/hg38/refGene/chr4:1,350,000-1,400,000
```

```
% curl http://togows.org/api/ucsc/hg38/refGene/inclusive/chr4:1,350,000-1,400,000
```

```
bin      name      chrom    strand  txStart  txEnd    cdsStart cdsEnd    exonCount exonStarts
      exonEnds score    name2    cdsStartStat    cdsEndStat    exonFrames
595      NM_020894 chr4      +        1347315  1388049  1348091  1385961  14
      1347315,1348089,1349523,1351714,1353029,1354734,1355116,1366319,1375363,1376033,1380046
,1380879,1383765,1385867,
      1347760,1348189,1349854,1351835,1353413,1354847,1355245,1366431,1375508,1376168,1380230
,1380988,1383940,1388049,    0      UVSSA    cpl     cpl     -1,0,2,0,1,1,0,0,1,2,2,0,1,2,
595      NM_001317934 chr4      +        1347265  1388049  1348091  1385961  14
      1347265,1348089,1349523,1351714,1353029,1354734,1355116,1366319,1375363,1376033,1380046
,1380879,1383765,1385867,
      1347609,1348189,1349854,1351835,1353413,1354847,1355245,1366431,1375508,1376168,1380230
,1380988,1383940,1388049,    0      UVSSA    cpl     cpl     -1,0,2,0,1,1,0,0,1,2,2,0,1,2,
595      NM_001317935 chr4      +        1347555  1388049  1348091  1385961  14
      1347555,1348089,1349523,1351714,1353029,1354734,1355116,1366319,1375363,1376033,1380046
,1380879,1383765,1385867,
      1347887,1348189,1349854,1351835,1353413,1354847,1355245,1366431,1375508,1376168,1380230
,1380988,1383940,1388049,    0      UVSSA    cpl     cpl     -1,0,2,0,1,1,0,0,1,2,2,0,1,2,
595      NM_175918 chr4      +        1391551  1395994  1394511  1395852  1      1391551,
      1395994,    0      CRIPAK  cpl     cpl     0,
```

Retrieve the refGene records fit within a region from 1,350,000 to 1,400,000bp on the chromosome 4 of the hg38 genome build.

```
% curl http://togows.org/api/ucsc/hg38/refGene/exclusive/chr4:1,350,000-1,400,000
```

```
bin      name      chrom    strand  txStart  txEnd    cdsStart cdsEnd    exonCount exonStarts
      exonEnds score    name2    cdsStartStat    cdsEndStat    exonFrames
595      NM_175918 chr4      +        1391551  1395994  1394511  1395852  1      1391551,
      1395994,    0      CRIPAK  cpl     cpl     0,
```

Show a summary of the wgEncodeBroadHistoneGm12878H3k27acStdSig dataset of the hg19 database.

```
% curl http://togows.org/api/ucsc/hg19/wgEncodeBroadHistoneGm12878H3k27acStdSig/info
```

```
version: 4
isCompressed: yes
isSwapped: 0
primaryDataSize: 198,894,024
primaryIndexSize: 1,440,088
zoomLevels: 10
chromCount: 23
basesCovered: 1,145,311,185
mean: 3.163029
min: 0.040000
max: 223899.000000
std: 98.594295
```

Retrieve the first 10 records from the wgEncodeBroadHistoneGm12878H3k27acStdSig dataset within a region from 1,000,000 to 2,000,000 on the chromosome 1 of the hg19 database.

```
% curl http://togows.org/api/ucsc/hg19/wgEncodeBroadHistoneGm12878H3k27acStdSig/chr1:1000000-2000000/10
2.87496 4.27916 5.23061 6.17385 4.07465 6.11871 9.60933 6.95731 3.53907 3.70842
```

Retrieve the genome sequence of a region from 12,345 to 12,500bp on the chromosome 1 of the hg38 database in a FASTA format.

```
% curl http://togows.org/api/ucsc/hg38/chr1:12,345-12,500.fasta
>hg38:chr1:12,345-12,500
TCAGACCAGCCGGCTGGAGGGAGGGGCTCAGCAGGTCTGGCTTTGGCCCTGGGAGAGCAG
GTGGAAGATCAGGCAGGCCATCGCTGCCACAGAACCCAGTGGATTGGCCTAGGTGGGATC
TCTGAGCTCAACAAGCCCTCTCTGGGTGGTAGGTGC
```

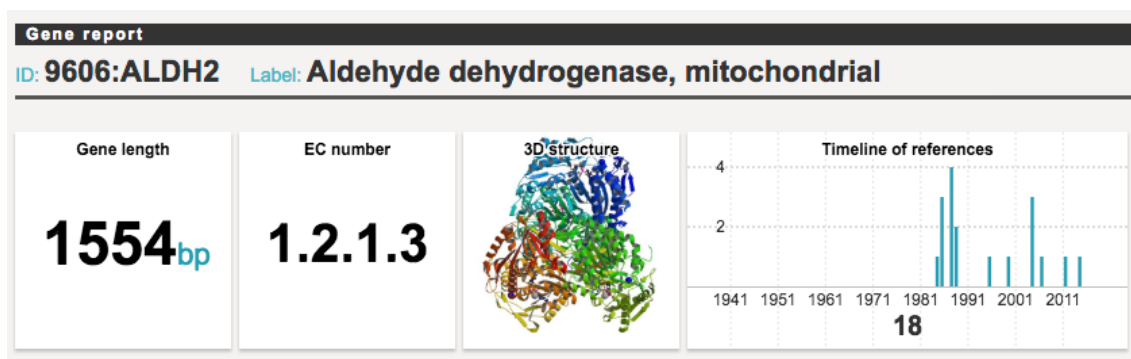
TogoStanza examples

In this section, representative examples of TogoStanza are shown.

TogoStanza in a gene report page

Summarized information of a gene report page in TogoStanza are shown, taking the human “ALDH2” gene as an example.

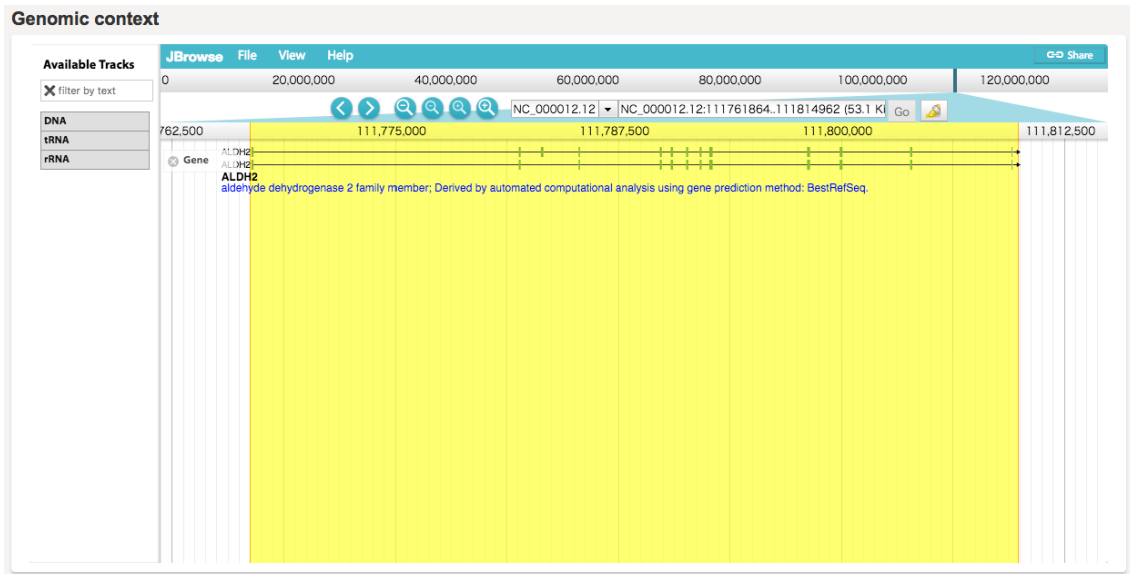
The gene report page of TogoGenome starts with a series of NanoStanza for “gene_length_nano”, “protein_ec_number_nano”, “protein3d_structure_nano” and “protein_references_timeline_nano”.



In the “Protein names” section, canonical and alternative names of a protein are shown by the “protein_names” stanza.

Protein names		
Protein names	Recommended Name	Aldehyde dehydrogenase, mitochondrial
	EC number	1.2.1.3
	Alternative Name(s)	<ul style="list-style-type: none">• ALDH class 2• ALDH-E2• ALDH1
Gene names	Name	<ul style="list-style-type: none">• ALDH2
	Synonyms	<ul style="list-style-type: none">• ALDM
Organism	Homo sapiens	
Taxonomic identifier	9606	

In the “Genomic context” section, gene structures of isoforms are shown by the “genome_jbrowse” stanza.



In the “Gene attributes” section, gene attributes such as a name, type, length and its location on the chromosome are shown by the “gene_attributes” stanza.

Gene attributes	
Gene name	ALDH2
Locus tag	
Gene type	CDS
Organism	Homo sapiens
RefSeq	NC_000012.12
Position	join(111766983..111767096,111781918..111782022,111783158..111783298,111785267..111785346,111789823..111789934,111790434..111790562,111791306..111791419,111792061..111792163,111792598..111792782,111798078..111798242,111799906..111800063,111803859..111803973,111809543..111809575)
Strand	Forward/positive strand position
Length	1554

[CSV](#) [JSON](#)

In the “Nucleotide sequence” section, the spliced nucleic acid sequence of a gene is shown by the “nucleotide_sequence” stanza.

Nucleotide sequence

```
ATGTTGCGCGCTGCGCCCGCTCGGGCCCGCTGGGCGCCGCTCTGTGAGCGCCGCCACCCAGGCGTGCTGCCCAACCAGCAGCCGAGGTCTTCTGCAACCAGATTTTCATAAACAATGAATGGCACGATGCCGTC
AGCAGGAAAACATTCACCACCGTCAATCCGTCACCTGGAGAGGTCACTGTGAGTAGCTGAAGGGGACAAGGAAGATGTGGACAAGGCAGTGAAGGCCGCCCGGGCCGCTTCCAGCTGGGCTCACCTTGGCGCCGATGGACGCA
TCACACAGGGCCGGCTGCTGAACCGCTGGCGATCTGATCGAGCGGGACCGGACCTCTGGCGGCTTGGAGACCTGGACAATGGCAAGCCCTATGTCACTCTACCTGGATTTGGACATGGTCTCAAAATGCTCCGG
TATTATGCGGCTGGGCTGATAAGTACACGGGAAAACATCCCATTTGACGGAGACTTCTTCACTACACACGCCATGAACCTGTGGGGTGTGCGGGCAGATCATCCGTGGAAATTCGCCCTCTGATGCAAGCATGGAAGCTG
GGCCAGCCTTGGCACTGGAACGTGGTTGTGATGAAGTAGCTGAGCAGACACCCCTCACCGCCCTCTATGTGGCAACCTGATCAAGGAGGCTGGCTTCCCCTGGTGTGGTCAACATTGTGCTGGATTTGGCCACAGGCT
GGGCGCCGCTTGCCTCCATGAGGATGTGGACAAAGTGGCATTACAGGCTCCACTGAGATTGGCCGCTAATCCAGGTTGCTGCTGGGAGCAGCAACCTCAAGAGAGTGAACCTGGAGCTGGGGGGGAAGAGCCCAACATCATC
ATGTGAGATGCCGATATGGATTGGCCGCTGGAACAGGCCCACTTCCGCTGTTCTTCAACAGGGCCAGTGCTGCTGTGCGGCTCCGGACCTTCTGTCAGGAGGACATCTATGATGAGTTTGTGGAGCGGAGCTTGGCCGGGCGC
AAGTCTCGGGTGTGGGAACCCCTTTGATAGCAAGACCAGCAGGGGGCCGAGTGGATGAACTCAGTTTAAAGAGATCTCGGCTACATCAACACGGGGAAGCAAGAGGGGGGGAAGCTGCTGTGTGGTGGGGCATTTGCTGCT
GACCGTGGTTACTTCACTCAGCCCACTGTGTTGGAGATGTGAGGATGGCATGACCATCGCAAGGAGGAGATCTCGGGCAGTGATGACAGATCTGAAGTTCAAGACATAGAGGAGGTTTGGGAGAGCCAAACAATCCAGC
TACGGGCTGGCCGAGCTGCTTCAACAAGGATTTGGACAAGGCCAATACCTGCTCCAGGCCCTCCAGCGGGCACTGTGGGTCAACTGCTATGATGTTTGGAGCCAGTCAACCTTTGGTGGCTACAAGATGTGGGGAGT
GGCCGGGAGTTGGCGAGTACGGCTGCAGGCATACACTGAAGTAAAACCTGTACAGTCAAAGTGCTCAGAAGAACTCATAA
```

[JSON](#)

In the “Protein attributes” section, the length of the amino acid sequence and the evidence of protein existence are shown by the “protein_attributes” stanza.

Protein attributes

Sequence length	517
Sequence status	Complete
Sequence processing	precursor
Protein existence	Evidence at protein level

[CSV](#) [JSON](#)

In the “Protein sequence” section, the amino acid sequence of a protein along with its length, molecular weight and ID is shown by the “protein_sequence” stanza.

Protein sequence

Sequence	Length	Mass (Da)	Status	Processing	Existence
P05091	517	56381	Complete	precursor	Evidence at protein level

Last modified: 1990-01-01, Version 2
Checksum: EBF74D44D285A00E

```
MLRAAARFGRLLSAAATQVVPAPNQPEVFNQIFINNEWHDAVSRKTFPTVNPSTGEVTCQVAEGKEDVDKAVKAARAAFQLGSPWRMDASHRGLLNRLADLIERDRTYLALETLDNGKPYYSYLVLDMLVCLKLR
YYAGWADKYHGKTIPTDGDFFSYTRHEPVGVCQIIPWNPFLMQAWKLGALATGNVVMKVAEQTLTALYVANLKEAGFPVGVNIVPFGFTAGAAIASHEDVDKVAFTGSTEIGRVIQVAAGSSNLKRVTLELGGKSPNII
MSDADMDIWAWEQAHFALFNQGGCCAGSRTFVQEDIYDFEVSARAKSRVGNPFDKTEQGPQVDETQFKLILGYINTKQEGAKLLCGGGIADRGYFIQPTVFGDVQDGMTIAKEEIFGPMQLKFKTIEEVVGRANNST
YGLAAAVFTKDLKXANLSQALQAGTWWNCYDVFAGQSPFGGYKMSGSGRELGEYGLQAYTEVKTVTVKVPQKNS
```

[CSV](#) [JSON](#)

In the “Protein general annotation” section, annotations of a subunit, similarity, polymorphism, caution, and subcellular location are shown by the “protein_general_annotation” stanza.

Protein general annotation	
Subunit	<ul style="list-style-type: none"> • Homotetramer.
Similarity	<ul style="list-style-type: none"> • Belongs to the aldehyde dehydrogenase family.
Polymorphism	<ul style="list-style-type: none"> • Genetic variation in ALDH2 is responsible for individual differences in responses to drinking alcohol [MIM:610251]. Allele ALDH2*2 is associated with a very high incidence of acute alcohol intoxication in Orientals and South American Indians, as compared to Caucasians.
Caution	<ul style="list-style-type: none"> • No experimental confirmation available.
Subcellular Location	<ul style="list-style-type: none"> • Mitochondrion matrix

[CSV](#) [JSON](#)

In the “Protein ontologies” section, keywords given by UniProt and annotations given by gene ontologies are shown by the “protein_ontologies” stanza.

Protein ontologies	
Keywords	
Cellular component	<ul style="list-style-type: none"> • Mitochondrion
Domain	<ul style="list-style-type: none"> • Transit peptide
Ligand	<ul style="list-style-type: none"> • Nicotinamide adenine dinucleotide, Nicotinic adenine dinucleotide, NAD
Molecular function	<ul style="list-style-type: none"> • Oxidoreductase
Technical term	<ul style="list-style-type: none"> • Reference proteome • 3D-structure • Direct protein sequencing
Gene Ontologies	
Biological process	<ul style="list-style-type: none"> • carbohydrate metabolic process • alcohol metabolic process • ethanol catabolic process • ethanol oxidation
Cellular component	<ul style="list-style-type: none"> • mitochondrial matrix • extracellular exosome
Molecular function	<ul style="list-style-type: none"> • aldehyde dehydrogenase (NAD) activity • aldehyde dehydrogenase [NAD(P)+] activity • electron carrier activity • glyceraldehyde-3-phosphate dehydrogenase (NAD+) (non-phosphorylating) activity • NAD binding

[CSV](#) [JSON](#)

In the “Protein sequence annotation” section, residue-based experimental information of a protein followed by modified residues, processing information, natural variations, secondary structures, frameshifts, and specific sites are shown by the “protein_sequence_annotation” stanza.

Protein sequence annotation

Experimental Information

Feature key	Position (s)	Length	Description	Feature identifier	Graphical view
Sequence Conflict	7-12	6	RFGPRL → ARAPP:		
Sequence Conflict	18	1	S → A:		
Sequence Conflict	60	1	S → F:		
Sequence Conflict	80-85	6	VKAARA → REGRPG:		
Sequence Conflict	101	1	R → S:		
Sequence Conflict	116	1	R → Q:		
Sequence Conflict	216	1	L → S:		
Sequence Conflict	218	1	A → R:		
Sequence Conflict	247	1	A → P:		
Sequence Conflict	332	1	Y → C:		
Sequence Conflict	362	1	V → L:		
Sequence Conflict	380	1	E → Q:		

Modification

Feature key	Position (s)	Length	Description	Feature identifier	Graphical view
Modified Residue	52	1	N6-acetyllysine		
Modified Residue	73	1	N6-acetyllysine		
Modified Residue	78	1	N6-acetyllysine		
Modified Residue	159	1	N6-acetyllysine		
Modified Residue	368	1	N6-acetyllysine		
Modified Residue	383	1	N6-acetyllysine		
Modified Residue	426	1	N6-acetyllysine		
Modified Residue	428	1	N6-acetyllysine		
Modified Residue	451	1	N6-acetyllysine		

Molecule Processing

Feature key	Position (s)	Length	Description	Feature identifier	Graphical view
Chain	18-517	500	Aldehyde dehydrogenase, mitochondrial	http://purl.uniprot.org/annotation/PRO_0000007168	
Transit Peptide	1-17	17	Mitochondrion		

Natural Variation

Feature key	Position (s)	Length	Description	Feature identifier	Graphical view
Sequence Variant	337	1	E → V:	http://purl.uniprot.org/annotation/VAR_011869	
Sequence Variant	496	1	E → K: In allele ALDH2*3.	http://purl.uniprot.org/annotation/VAR_011302	
Sequence Variant	504	1	E → K: In allele ALDH2*2; drastic reduction of enzyme activity.	http://purl.uniprot.org/annotation/VAR_002248	
Splice Variant	74-120	47	EDVDKAVKAAARAAFLGSPWRRMDASHRGLLRNRLADLIERDRTYLA → : In isoform 2.	http://purl.uniprot.org/annotation/VSP_046715	

Region

Feature key	Position (s)	Length	Description	Feature identifier	Graphical view
Nucleotide Phosphate Binding	262-267	6	NAD		

Secondary Structure						
Feature key	Position (s)	Length	Description	Feature identifier		Graphical view
Helix	73-86	14				
Helix	92-95	4				
Helix	98-114	17				
Helix	116-127	12				
Helix	131-136	6				
Helix	138-152	15				
Helix	188-201	14				
Helix	216-228	13				
Helix	245-250	6				
Helix	264-276	13				
Helix	300-312	13				
Helix	313-316	4				
Helix	329-345	17				
Helix	364-379	16				
Helix	411-414	4				
Helix	430-438	9				
Helix	453-462	10				
Helix	486-488	3				
Helix	496-502	7				
Strand	38-41	4				
Strand	44-46	3				
Strand	53-57	5				
Strand	64-69	6				
Strand	158-161	4				
Strand	164-175	12				
Strand	178-182	5				
Strand	185-187	3				
Strand	205-209	5				
Strand	212-214	3				
Strand	234-237	4				
Strand	257-262	6				
Strand	281-285	5				
Strand	290-294	5				
Strand	322-328	7				
Strand	383-386	4				
Strand	389-391	3				
Strand	393-396	4				
Strand	401-405	5				
Strand	419-427	9				
Strand	439-441	3				
Strand	444-449	6				
Strand	465-471	7				
Strand	478-480	3				
Strand	489-491	3				
Strand	503-511	9				
Turn	59-61	3				
Turn	153-155	3				
Turn	242-244	3				

Sequence Caution					
Feature key	Position (s)	Length	Description	Feature identifier	Graphical view
Frameshift	424	1			
Frameshift	444	1			
Frameshift	448	1			
Frameshift	461	1			

Site					
Feature key	Position (s)	Length	Description	Feature identifier	Graphical view
Active Site	285	1	Proton acceptor		
Active Site	319	1	Nucleophile		
Site	186	1	Transition state stabilizer		

[CSV](#) [JSON](#)

In the “Protein orthologs” section, UniProt IDs of orthologous proteins are shown by the “protein_orthologs” stanza.

Protein orthologs

Q4WVW0, Q4W9H6, Q4WA32, Q4W9U5, Q4WP63, Q4WPA5, Q4WQH7, Q4WQP1, Q4WD76, Q4WMT8, Q4WM26, Q4WAE3, Q4WBG0, Q4WBH5, Q4WC77, Q4WC68, D2RNG0, C7M3D3, C7LYS3, C7LZH6, C7LZZ6, Q75E29, Q75F03, Q75CG3, F2G3K2, F2GBM3, D8HT20, D8IOU7, D8I298, D8I299, D8I5F6, D8I5G2, D8I8B6, D8HPJ0, D8HWC3, D8HXX0, D8HXX3, D8HY14, D8I2K1, D8I4C0, D8HNMO, D8HNMM7, D8HQY5, D8HS52, D9PZ41, F6ER14, F6EL78, F6EM37, F6EPL2, F6ERN1, F6EG88, F6EJ78, F6EP03, F6EQI9, F6ER27, F6ES16, F6EF18, F6EFN1, F6EGK1, F6EHC9, F6E158, F6E181, F6EIC6, F6EKK4, F6EKY2, F6ES21, F6E541, F6EHP5, F6ELO8, G8SLA0, G8RZF7, G8RZF9, G8S0L8, G8S3I4, B9DHD2, F4INS6, Q1WIQ6, Q56YU0, Q95U63, Q95T51, A1K6L4, A1K6T2, A1K7D1, A1K7R2, A1K897, A1K8D5, A1K9G3, A1K9Q0, A1K9Y2, A1KC13, Q6MRF6, Q6MQ98, Q6MNM1, Q6MNT9, Q6MLS9, Q6MKW1, I3Z0H4, I3Z180, I3Z440, I3Z9V3, I3ZAS0, C0Z4D9, C0ZB85, A0JQW7, A0JRG4, A0JRU1, A0JS01, A0JTS1, A0JTV0, A0JU81, A0JUJ7, A0JVP7, A0JV S7, A0JVV6, A0JW23, A0JW58, A0JWA6, A0JWG2, A0JXH3, A0JZJ7, A0JZK4, A0K090, A0K0R6, A0K057, A0K0W6, A0K0Z7, A0K1D0, A0K1E9, A0K1T9, D9Q1G1, Q8A340, M4 RBL7, M4R6F2, D5WT37, D5WTT0, D5WVH7, D5WVW5, D5WW73, D5WUW9, D5WXP7, D5WXV6, D5WPK8, Q8VZC3, E8MZ95, H6LKZ1, K0IZM4, E3HT37, E3HT68, E3HLU2, E3HTE8, E3HHR5, E3HNI3, E3HPU7, M5A698, M5A6A6, A8IG36, A8IUK7, A8IML5, A8IQB9, A8HRG6, A8I4Z1, A8I2Q5, A8I6R8, A8I6R9, A8I8B2, A8IDA9, A8IKT5, A8HRZ8, A8 HSF5, D3NVL7, D3P0S6, D3P196, D3P1E4, D3P1W1, D3P3M7, D3P2C0, D3P2K3, D3P2L2, D3P3B2, D3P3U1, D3P3Z6, D3P4N4, D3P4N6, D3P634, D3P6D6, D3P6M 6, D3P754, D3P758, D3P7X9, A1K1X7, A1K1Y4, Q7WRB4, Q7WQK8, Q7WPP3, Q7WPN3, Q7WPK7, Q7WPE8, Q7WPE0, Q7WP17, Q7WNH1, Q7WN24, Q7WMB1, Q7WM30, Q 7WLE3, Q7WL74, Q7WKT0, Q7WJZ0, Q7WJV4, Q7WJ53, Q7WIB5, Q7WHS3, Q7WGM8, Q7WD46, Q7WFF4, Q7WEJ2, Q7WDM3, C5BVN4, C5BVN6, C5BYN2, C5BYN8, C5BYT9

In the “Protein references” section, a list of related literature is shown by the “protein_references” stanza.

Protein references

An enzyme assisted RP-RPLC approach for in-depth analysis of human liver phosphoproteome.
Huang J, Sun D, Wang F, Wang L, Song C, Ye M, Zou H, Bian Y, Dong M, Cheng K.
J. Proteomics 96 253-262 (2014-01-01T00:00:00+09:00) <https://www.ncbi.nlm.nih.gov/pubmed/24275569>

Initial characterization of the human central proteome.
Bennett K.L., Superti-Furga G., Colinge J., Kaupé I., Burkard T.R., Planyavsky M., Breitwieser F.P., Buerckstuemmmer T.
BMC Syst. Biol. 5 17 (2011-01-01T00:00:00+09:00) <https://www.ncbi.nlm.nih.gov/pubmed/21269460>

The finished DNA sequence of human chromosome 12.
Chen G., Chen R., Chen Z., Harris R.A., Hernandez J., Huang M., Johnson R., Kucherlapati R., Gibbs R.A., Lee S., Li Z., Li L., Liu J., Liu W., Lu J., Ma J., Loulseged H., Martinez E., Mitchell T., Morris S., Perez A., Shen H., Wang L., Wang Q., Wei X., Wheeler D.A., Williamson A.L., Zhang J., Zhang Z., Zhou J., Bailey M., Brown M.J., Davis C., Davis C.M., Jones S., Jolivet A., Mu

In the “Protein cross references” section, links to other entries of organism-specific databases, PTM databases, phylogenomic databases, sequence databases, genome annotation databases, 2D gel databases, family and domain databases, protein-protein

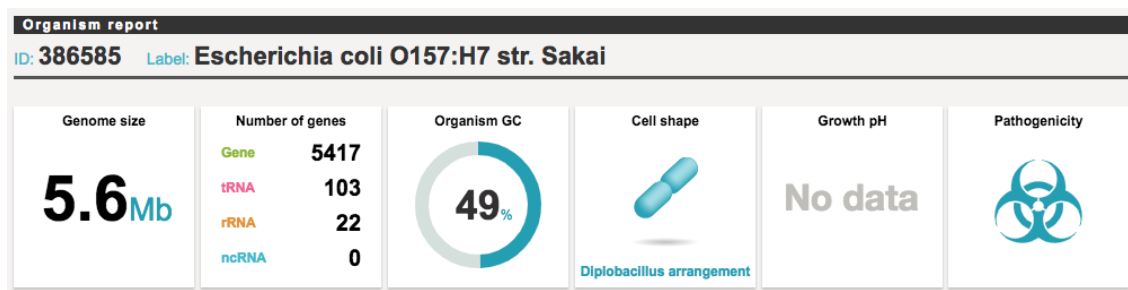
interaction databases, 3D structure databases, enzyme and pathway databases, proteomic databases, gene expression databases, and other databases are shown by the “protein_cross_references” stanza.

Protein cross references	
Organism-specific databases	
neXtProt	• NX_P05091
PharmGKB	• PA24696
OpenTargets	• ENSG00000111275
MalaCards	• ALDH2
MIM	• 610251 • 100650
HPA	• HPA051065
HGNC	• 404
GeneCards	• ALDH2
EuPathDB	• HostDB:ENSG00000111275.12
DisGeNET	• 217
CTD	• 217
PTM databases	
iPTMnet	• P05091
SwissPalm	• P05091
PhosphoSitePlus	• P05091
Phylogenomic databases	
eggNOG	• KOG2450 • COG1012
TreeFam	• TF300455
PhylomeDB	• P05091
OrthoDB	• EOG091G05E8
OMA	• TFVQEDV
KO	• K00128
InParanoid	• P05091
HOVERGEN	• HBG000097
HOGENOM	• HOG000271505
GeneTree	• ENSGT00940000156240

TogoStanza in an organism report page

Summarized information of an organism report page in TogoStanza are shown, taking the “*Escherichia coli* O157” as an example.

The organism report page of TogoGenome starts with a series of NanoStanza for “organism_genome_size_nano”, “organism_gene_number_nano”, “organism_gc_nano”, “organism_microbial_cell_shape_nano”, “organism_ph_nano”, and “organism_related_disease_nano”.



In the “Organism name” section, scientific and alternative names of an organism are shown by the “organism_names” stanza.

The figure shows the “Organism name” section with scientific and equivalent names for *Escherichia coli* O157:H7 str. Sakai.

Organism name	
Scientific name	• <i>Escherichia coli</i> O157:H7 str. Sakai
Equivalent name	• <i>Escherichia coli</i> O157:H7 strain Sakai

CSV JSON

In the “Genome information” section, chromosomes and organelle genomes with statistics and links are shown by the “genome_information” stanza.

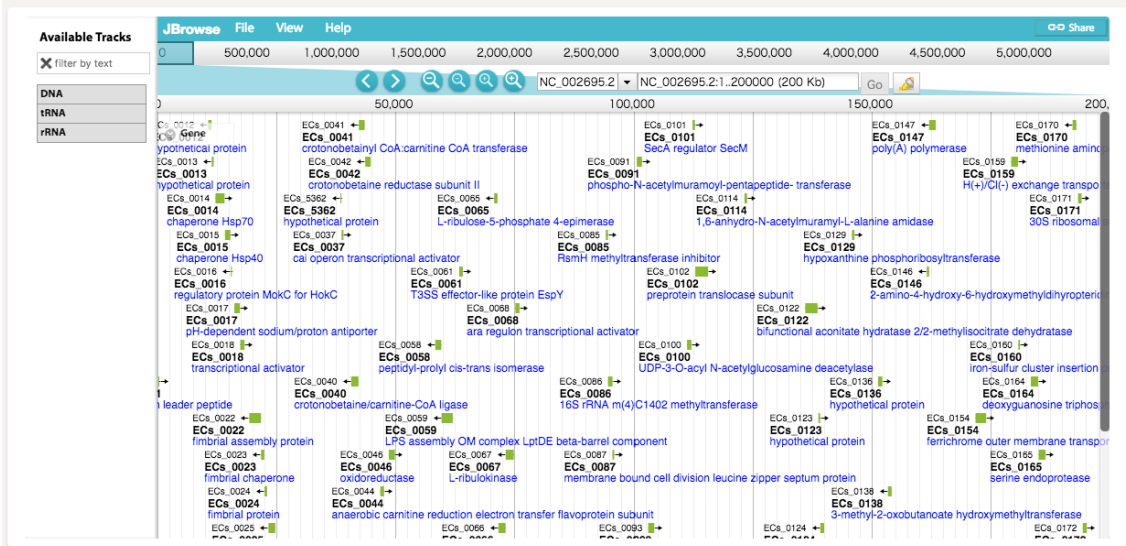
Genome information

ASM886v2							
Description	RefSeq	Type	Size	Gene	tRNA	rRNA	Other
Escherichia coli O157:H7 str. Sakai DNA, complete genome.	NC_002695.2	chromosome	5498578	5329	103	22	90
Escherichia coli O157:H7 str. Sakai plasmid pO157, complete sequence.	NC_002128.1	plasmid	92721	85	0	0	4
Escherichia coli O157:H7 str. Sakai plasmid pOSAK1, complete sequence.	NC_002127.1	plasmid	3306	3	0	0	1

[CSV](#) [JSON](#)

In the “Genomic context” section, genes on the genome browser are shown by the “genome_jbrowse” stanza.

Genomic context



In the “Ortholog profile” section, the taxonomic profile of orthologous gene groups is shown by the “taxonomy_ortholog_profile” stanza (this example is taken from “*E. coli* str. K-12 substr. MG1655”).

Ortholog profile

29	Glutamate synthase / formate dehydrogenase synthase beta subunit / formate dehydrogenase alpha	29
22	Transposase, IS4 family protein	22
22	Amino acid transporter permease	22
22	Amino acid transporter permease	22
21	Alcohol dehydrogenase, zinc protein	21
21	Integrase/transposase	21
20	RHS repeat-associated core domain-containing protein	20
19	Integrase/transposase	19
19	Binding transcriptional regulator LacI family protein	19
17	4Fe-4S ferredoxin iron-sulfur binding-containing protein	17

[JSON](#) [SVG](#)

In the “Taxonomic information” section, the taxonomic lineage of an organism is shown by the “lineage_information” stanza.

Taxonomic information

Rank	Lineage	Taxonomy ID
	cellular organisms	131567
	Bacteria	2
	Proteobacteria	1224
	Gammaproteobacteria	1236
	Enterobacteriales	91347
	Enterobacteriaceae	543
	Escherichia	561
	Escherichia coli	562
	Escherichia coli O157:H7	83334
	Escherichia coli O157:H7 str. Sakai	386585

[CSV](#) [JSON](#)

In the “Culture collections” section, the related strains of an organism are shown by the “organism_culture_collections” stanza (this example is taken from “*Nocardia higoensis*”).

Culture collections

Strain No.	Organism name	Isolation	Environmental link	Type strain	Applications	Other collections
NBRC 0001	Rhodotorula mucilaginosa (Jörgensen) Harrison			No		CBS 328, MUCL 30593, VKM Y-7
JCM 12121	Nocardia higoensis Kageyama et al.	Patient with pleurisy, Japan		Yes		CIP 108597, DSM 44732, IFM 10084, NBRC 100133
NBRC 100133	Nocardia higoensis Kageyama et al. 2004	Human		Yes		CIP 108597, DSM 44732, IFM 10084, JCM 12121

[CSV](#) [JSON](#)

In the “Medium information” section, the medium of an organism is shown by the “organism_medium_information” stanza (this example is taken from “*Nocardia higoensis*”).

Medium information

Medium ID	NBRC_M229
Medium name	
Medium type	Undefined medium
Defined components	Calcium carbonate, Glycerol
Undefined components	Yeast extract
Solidifying components	Agar
Water	Distilled water

[CSV](#) [JSON](#)

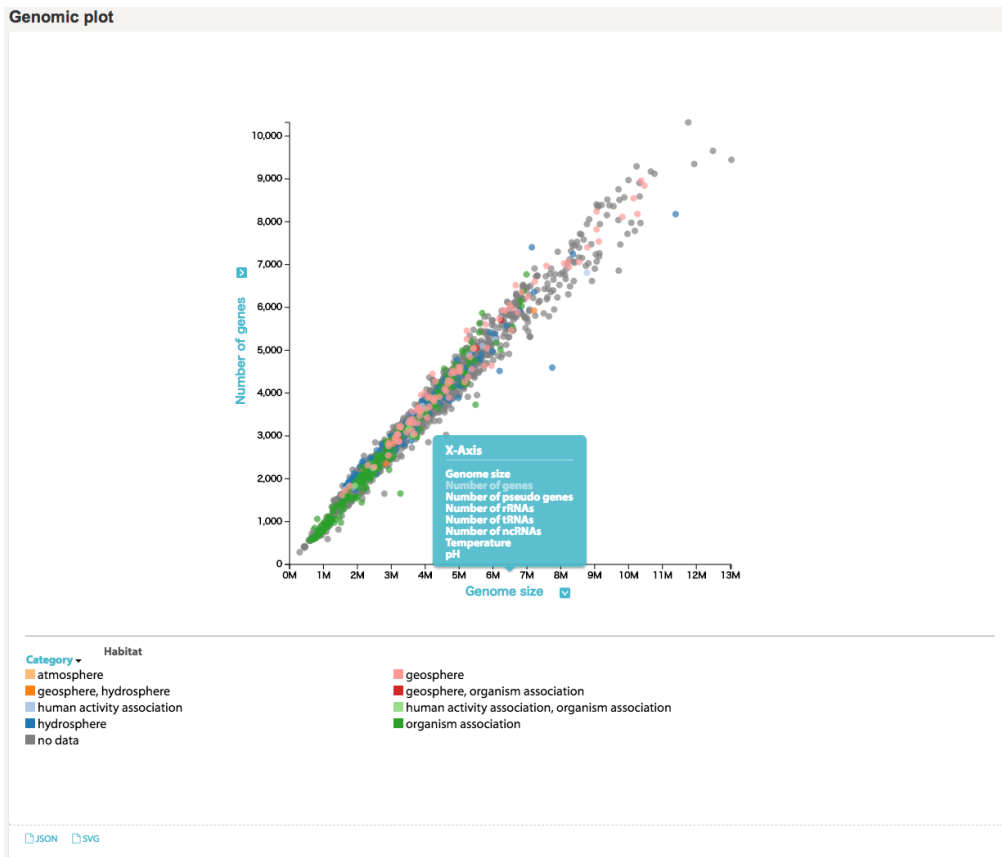
In the “Phenotype information” section, the phenotypic features of an organism are shown by the “organism_phenotype” stanza.

Phenotype information

Show cell shape	Rod shape
Oxygen requirement	Facultative anaerobe
Temperature range	Mesophilic
Optimal growth temperature	37°C
Show motility	Motile
Show cell arrangement	Single arrangement, Pair arrangement
Representative morphology	Diplobacillus arrangement
Staining	Gram negative
Show morphology	Rod shape, Diplobacillus arrangement

[CSV](#) [JSON](#)

In the “Genomic plot” section, a scatter plot by selected features of organisms is shown by the “genome_plot” stanza.



In the “Pathogen information” section, infectious diseases of an organism are shown by the “organism_pathogen_information” stanza.

Pathogen information

Organism name	Disease name	Infectious type	Strain type
Coxiella burnetii RSA 331	Food poisoning		
Escherichia coli O157:H7 str. Sakai	Food poisoning		
Salmonella enterica subsp. enterica serovar Typhimurium str. SL1344	Food poisoning		
Shigella sonnei 53G	Food poisoning		
Listeria monocytogenes EGD-e	Food poisoning		

CSV JSON

In the “Organism cross references” section, cross references to other databases are shown by the “organism_cross_references” stanza.

Organism cross references	
GOLD	Gc00046 , Gi0047830
RefSeq	NC_002127 , NC_002127.1 , NC_002128 , NC_002128.1 , NC_002695 , NC_002695.2
CSV JSON	

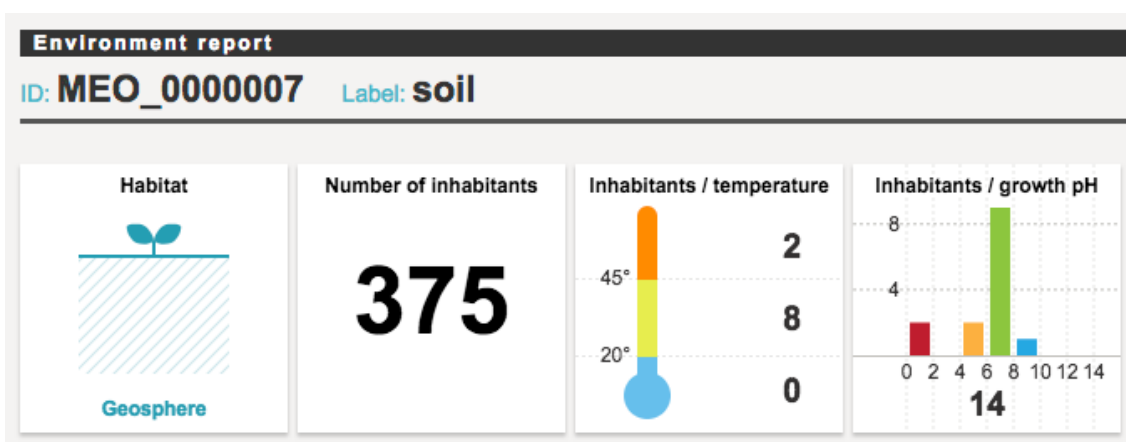
In the “Genome cross references” section, cross references to genome databases are shown by the “genome_cross_references” stanza.

Genome cross references	
Assembly:ASM886v2	
NC_002127.1 : Escherichia coli O157:H7 str. Sakai plasmid pOSAK1, complete sequence.	
BioProject	PRJNA57781
RefSeq	NC_002127.1
NC_002128.1 : Escherichia coli O157:H7 str. Sakai plasmid pO157, complete sequence.	
BioProject	PRJNA57781
RefSeq	NC_002128.1
NC_002695.2 : Escherichia coli O157:H7 str. Sakai DNA, complete genome.	
BioProject	PRJNA226
BioSample	SAMN01911278
RefSeq	NC_002695.2
CSV JSON	

TogoStanza in an environment report page

Summarized information of an environment report page in TogoStanza are shown, taking “soil” as an example.

The environment report page of TogoGenome starts with a series of NanoStanza for “environment_top_level_symbolic_image_nano”, “environment_inhabitants_statistics_nano”, “environment_organism_distribution_on_temperature_nano”, and “environment_organism_distribution_on_ph_nano”.



In the “Environment attributes” section, the name and description of an environment are shown by the “environment_attributes” stanza.

Environment attributes	
Environment	soil
Description	Any material within 2 m from the Earth's surface that is in contact with the atmosphere, with the exclusion of living organisms, areas with continuous ice not covered by other material, and water bodies deeper than 2 m.
MEO	MEO_0000007
Exact synonyms	
<input type="checkbox"/> CSV <input type="checkbox"/> JSON	

In the “Inhabitants statistics” section, statistics of organisms in an environment are shown by the “environment_inhabitants_statistics” stanza.

Inhabitants statistics

GOLD	375
JCM	2569
NBRC	3279

[CSV](#) [JSON](#)

In the “Inhabitants” section, a list of inhabitants is shown by the “environment_inhabitants” stanza.

Inhabitants

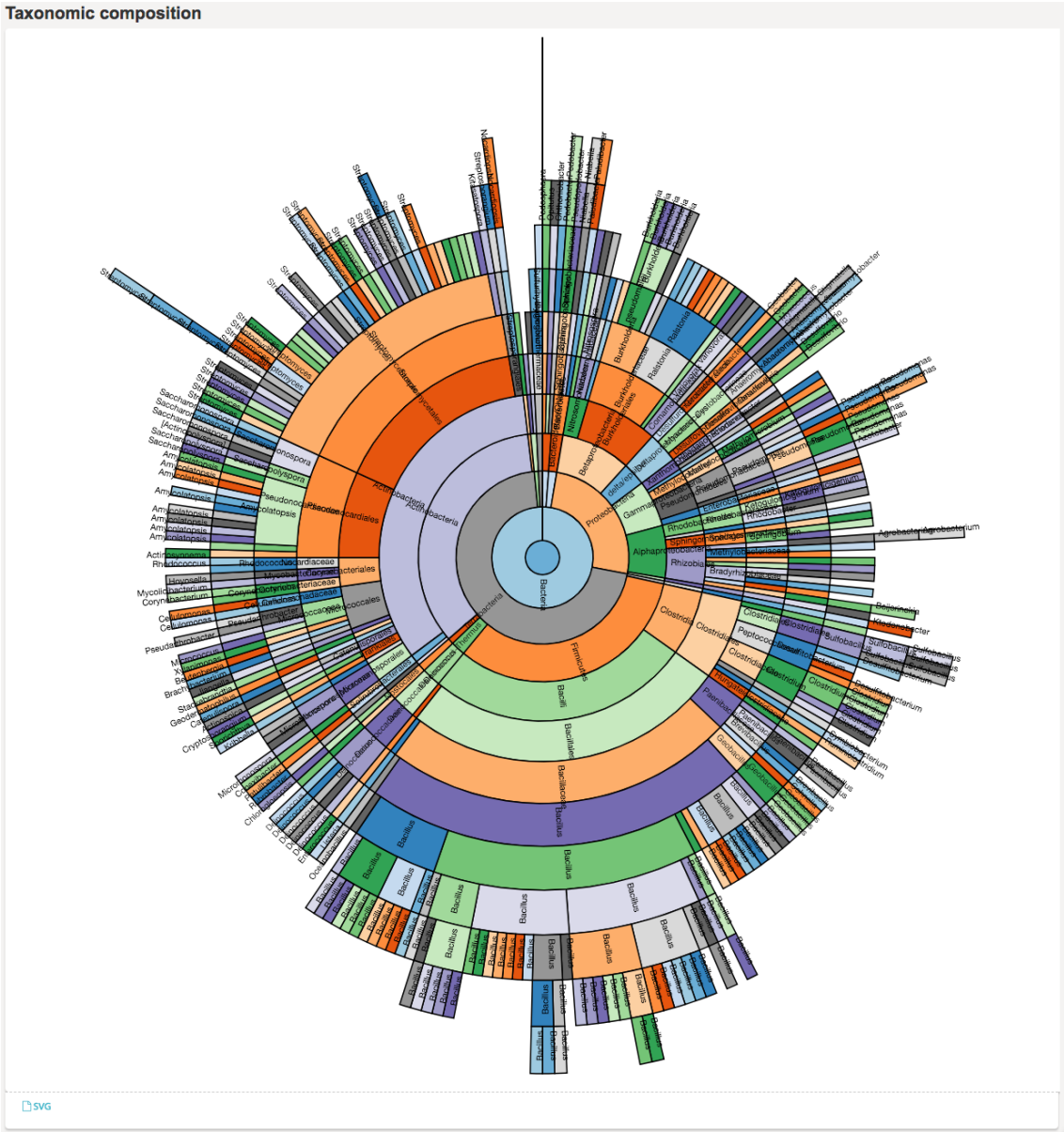
Original source	Organism name	Taxonomy ID	Isolation	Environments
GI02991				soil
GI09171				soil
Gc01728				soil
Gc00166				soil
GI01834				soil
Gc01971				soil
Gc00932				soil
GI05549				soil
GI04162				soil
GI03390				soil
Gc01041				soil

In the “Geographical map” section, geographical locations of inhabitants are shown by the “environment_geographical_map” stanza.

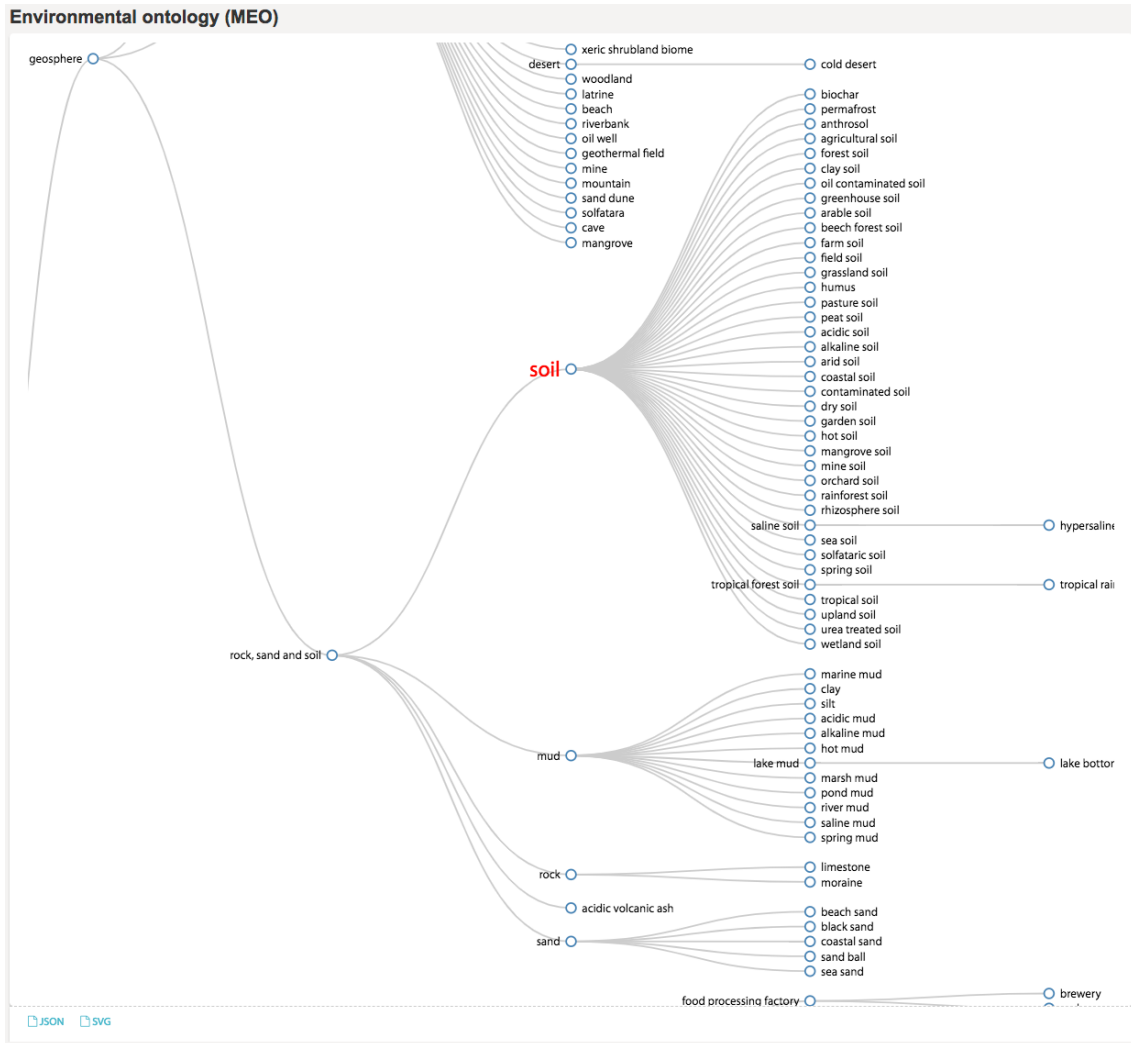
Geographical map



In the “Taxonomic composition” section, the taxonomic composition of inhabitants is shown by the “environment_taxonomic_composition” stanza.



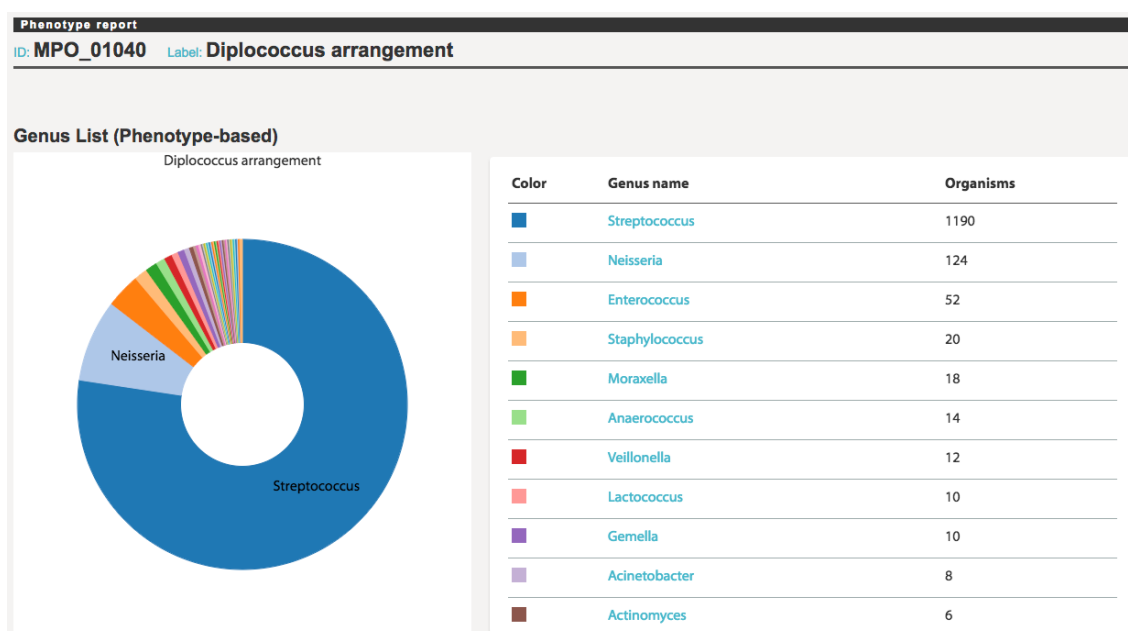
In the “Environmental ontology (MEO)” section, the hierarchical classification of the MEO environmental ontology is shown by the “environment_environmental_ontology” stanza.



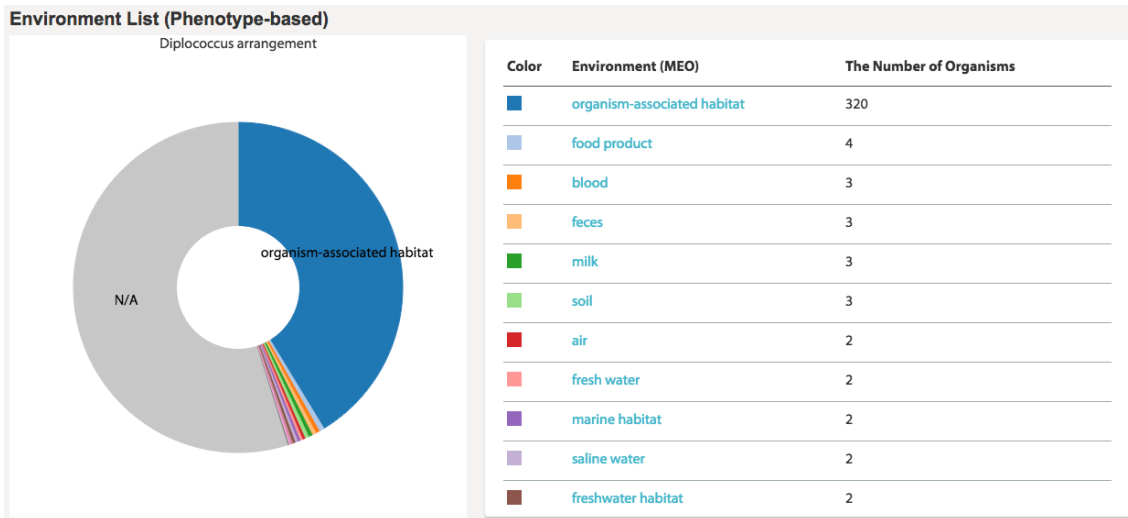
TogoStanza in a phenotype report page

Summarized information of a phenotype report page in TogoStanza are shown, taking “Diplococcus arrangement” as an example.

In the “Genus list” section, statistics and a list of species having the same phenotype are shown by the “microbial_phenotype_genus_composition” stanza.



In the “Environment list” section, statistics and a list of environments where inhabitants having the same phenotype are shown by the “microbial_phenotype_environment_composition” stanza.



In the “Shape information” section, a brief description of a phenotype is shown by the “microbial_phenotype_cell_shape” stanza.

Shape Information

Diplococcus arrangement

Synonym

- Diplococcus

Definition
(No Data)

Synonyms can also be shown, as in the case of “Rod shape”

Shape Information

Rod shape

Synonym

- Bacillus
- Rod
- Rod-shaped

Definition
(No Data)

In the “Organism list” section, a list of organisms having the same shape is shown by the “microbial_phenotype_information” stanza.

Organism List		
Acinetobacter		
Organism name	Taxonomy ID	Phenotype
Acinetobacter baumannii 6014059	525242	Diplococcus arrangement, Diplococcus arrangement
Acinetobacter baumannii 6013113	592014	Diplococcus arrangement, Diplococcus arrangement
Acinetobacter baumannii BJAB07104	1096995	Diplococcus arrangement, Diplococcus arrangement
Acinetobacter sp. ATCC 27244	525244	Diplococcus arrangement, Diplococcus arrangement
Actinomyces		
Organism name	Taxonomy ID	Phenotype
Actinomyces cardiffensis F0333	888050	Diplococcus arrangement, Diplococcus arrangement
Actinomyces sp. oral taxon 178 str. F0338	888051	Diplococcus arrangement, Diplococcus arrangement
Actinomyces sp. oral taxon 180 str. F0310	888052	Diplococcus arrangement, Diplococcus arrangement
Aerococcus		
Organism name	Taxonomy ID	Phenotype
Aerococcus urinae ACS-120-V-Col10a	866775	Diplococcus arrangement, Diplococcus arrangement
Aerococcus viridans ATCC 11563 = CCUG 4311	655812	Diplococcus arrangement, Diplococcus arrangement
Akkermansia		
Organism name	Taxonomy ID	Phenotype
Akkermansia muciniphila ATCC BAA-835	349741	Diplococcus arrangement, Diplococcus arrangement
Alloscardovia		
Organism name	Taxonomy ID	Phenotype
Alloscardovia omnicolens F0580	1321816	Diplococcus arrangement, Diplococcus arrangement
Anaerococcus		
Organism name	Taxonomy ID	Phenotype
Anaerococcus hydrogenalis DSM 7454	561177	Diplococcus arrangement, Diplococcus arrangement
Anaerococcus hydrogenalis ACS-025-V-Sch4	879306	Diplococcus arrangement, Diplococcus arrangement
Anaerococcus lactolyticus ATCC 51172	525254	Diplococcus arrangement, Diplococcus arrangement
Anaerococcus prevotii DSM 20548	525919	Diplococcus arrangement, Diplococcus arrangement
Anaerococcus prevotii ACS-065-V-Col13	879305	Diplococcus arrangement, Diplococcus arrangement
Anaerococcus tetradius ATCC 35098	525255	Diplococcus arrangement, Diplococcus arrangement
Anaerococcus vaginalis ATCC 51170	655811	Diplococcus arrangement, Diplococcus arrangement