SPINing quality into the Semantic Web: a biocuration case study

Jerven Bolleman[1], Sebastien Gehant[1] and the UniProt Consortium[1,2,3]

[1]Swiss-Prot Group, SIB Swiss Institute of Bioinformatics, rue Michel-Servet 1, 1211 Geneva 4, Switzerland
[2]European Bioinformatics Institute (EBI), Welcome Trust Genome Campus, Hinxton Cambridge CB101SD UK
[3]Protein Information Resource (PIR), Georgetown University Medical Center, 3300 Whitehaven Street, NW, Suite 1200, Washington, DC 20007, USA

UniProt a "Knowledgebase"

The UniProt Knowledgebase (UniProtKB)[1] consists of two sections:

- UniProtKB/Swiss-Prot contains manually reviewed records with annotation extracted from the literature and curator evaluated computational analysis.
- UniProtKB/TrEMBL contains computationally generated records enhanced by automatic classification and annotation.

One of the central activities of the UniProt Consortium is the biocuration of UniProtKB/Swiss-Prot. Providing high quality and consistent annotation depends on curation experts who apply common curation standards. Customized curation tools help to apply standards and to avoid trivial mistakes, thereby allowing curators to focus on the scientific content.

We propose to use SPIN to encode and apply rules that ensure curation standards.

SPIN

SPARQL Inferencing Notation (SPIN)[2] is a small set of technologies, built on the SPARQL standard, that is aimed at making reasoning and data validation easy.

SPIN is suitable for checking both schema compliance and data integrity. For UniProtKB we validate that entries conform to our OWL "schema" and we check our own custom curation rules.

We maintain the SPIN rules with the integrated development environment "TopBraid composer". The rules are documented on an internal website and are searchable with SPARQL.

Example 1: Atypical homeoboxes don't bind DNA.

```
CONSTRUCT
  { _:c0 rdf:type spin:ConstraintViolation .
    _:c0 spin:violationRoot keyword:371 .
    _:c0 spin:violationPath core:classifiedWith .
    _:c0 rdfs:label "URC90: If keyword:\"Homeobox[KW-0371]\" is
present we expect keyword:\"DNA-binding[KW-0238]\"" .}
WHERE
  { ?this rdf:type core:Protein .
    ?this core:classifiedWith keyword:371
    NOT EXISTS
      { ?this core:classifiedWith keyword:238 }
    NOT EXISTS
      { ?this core:annotation ?annotation .
        ?annotation rdf:type core:Nucleotide_Binding_Annotation .
        ?annotation rdfs:comment ?comment .
        ?this core:annotation ?functionalAnnotation .
        ?functionalAnnotation rdf:type core:Function_Annotation .
        ?functionalAnnotation rdfs:comment ?functionalComment
        FILTER ( regex(?comment, "Homeobox; atypical") && regex(?
functionalComment, "does not bind DNA") )
      }
  }
```

Violation triples

We build 4 triples whenever a constraint is violated to make the violation accessible as an object which describes the violating resource (via :violationRoot and :violationPath) and provides a human readable error message. The violating resource belongs to a (not shown) named graph which links the resource with the appropriate GUI element of the curation platform. This link is used to display the error message next to the data that triggered the error, in this example, the presence of the "Homeobox" keyword without its expected complement "DNA-binding".

Finding "bad" data

SPARQL works by defining graph patterns to match. In this example, we are looking for a "Protein" (UniProtKB entry) that is classified with the keyword "Homeobox", while excluding the entries which also have the keyword "DNA-binding" as those are correct.

Avoiding false positives

There are a few proteins with atypical homeobox patterns that do not bind DNA and we must not raise an error for these. Instead of hard coding such exceptions, we use whenever possible information in the record to avoid false positives.

Performance

SPARQL is considered to be a slow technology, but SPIN rules are trivial to run in parallel due to rule independence. In this case study, it took 23 hours to check all 21 million UniProtKB entries against 137 rules on a 4 CPU Intel X7350 64 GB RAM machine. The 0.5 million UniProtKB/Swiss-Prot entries can be processed within 3 hours on a 2 CPU virtual machine. It takes on average 1/4 of a millisecond to check one rule against one entry.

Powered by Vital-IT

Rules for rules

Maintaining rules is as important as implementing them. As both our insight into biology and our database change, we need to adapt our rules accordingly.

Since SPIN rules are encoded in RDF, like our data, we can easily make SPIN rules on SPIN rules. Example 2 shows a rule that checks that all rules contain only valid taxonomy identifiers. Comparable checks would be hard to express in languages where data is encoded differently from code.

Example 2: Rule checking for obsolete taxonomy identifiers in other rules.

```
PREFIX spin:<http://spinrdf.org/spin#>
PREFIX sp:<http://spinrdf.org/sp#>
PREFIX up:<http://purl.uniprot.org/core/>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>

CONSTRUCT {
   _:b0 a spin:ConstraintViolation .
   _:b0 spin:violationRoot ?this .
   _:b0 spin:violationPath sp:Construct .
   _:b0 rdfs:label "Rule should not mention an obsolete taxon." .
} WHERE {
   ?this spin:constraint ?constraint .
   ?constraint a sp:Construct .
   #Find the where clause in another rule.
   ?constraint sp:where ?list .
   #property expression to find all elements in a list.
   ?list rdf:first|((rdf:rest/rdf:first))* ?item .
   ?item sp:predicate up:organism .
   ?item sp:object ?object .
   ?object a up:Taxon .
   ?object up:obsolete true .
}
```

Further work

The SPIN rules must be integrated into the C++ based curation platform. SPIN violations must be mapped to GUI elements to show the errors in the context in which they occur to make it easy for curators to fix them.

We also need to validate that the SPIN rules cover at least all rules that are implemented by the legacy Perl based system.

1. Reorganizing the protein space at the Universal Protein Resource (UniProt) Nucleic Acids Res. 40: D71-D75 (2012). http://dx.doi.org/doi:10.1093/nar/gkr981.
2. W3C: SPIN - Overview and Motivation 2011, http://www.w3c.org/Submission/2011/SUBM-spin-overview-20110222

UniProt is mainly supported by the National Institutes of Health (NIH) grant 1 U41 HG006104. Additional support for the EBI's involvement in UniProt comes from the NIH grant 2P41 HG02273. Swiss-Prot activities at the SIB are supported by the Swiss Federal Government through the Federal Office of Education and Science and the European Commission contracts SLING (226073), Gen2Phen (200754) and MICROME (222886). PIR's UniProt activities are also supported by the NIH grants 5R01GM080646-07, 3R01GM080646-07S1, 5G08LM010720-03, and 8P20GM103446-12, and the National Science Foundation (NSF) grant DBI-1062520.

Contact
help@uniprot.org
www.uniprot.org

See also:
www.isb-sib.ch