GigaScience, 9, 2020, 1–7

doi: 10.1093/gigascience/giaa003
Research

RESEARCH

HAMAP as SPARQL rules—A portable annotation pipeline for genomes and proteomes

Jerven Bolleman [1,*], Edouard de Castro [1], Delphine Baratin [1], Sebastien Gehant [1], Beatrice A. Cuche [1], Andrea H. Auchincloss [1], Elisabeth Coudert [1], Chantal Hulo [1], Patrick Masson [1], Ivo Pedruzzi [1], Catherine Rivoire [1], Ioannis Xenarios [1,2], Nicole Redaschi [1] and Alan Bridge [1]

[1] Swiss-Prot Group, SIB Swiss Institute of Bioinformatics, Centre Médical Universitaire, 1 rue Michel-Servet, CH-1211 Geneva 4, Switzerland; and [2] Centre Hospitalier Universitaire Vaudois/Ludwig Institute for Cancer Research, Agora Centre, CH-1005 Lausanne, Switzerland

[*] **Correspondence address.** Jerven Bolleman, Swiss-Prot Group, SIB Swiss Institute of Bioinformatics, Centre Médical Universitaire, 1 rue Michel-Servet, CH-1211 Geneva 4, Switzerland. E-mail: jerven.bolleman@sib.swiss http://orcid.org/0000-0002-7449-1266, hamap@sib.swiss

Abstract

Background: Genome and proteome annotation pipelines are generally custom built and not easily reusable by other groups. This leads to duplication of effort, increased costs, and suboptimal annotation quality. One way to address these issues is to encourage the adoption of annotation standards and technological solutions that enable the sharing of biological knowledge and tools for genome and proteome annotation. **Results:** Here we demonstrate one approach to generate portable genome and proteome annotation pipelines that users can run without recourse to custom software. This proof of concept uses our own rule-based annotation pipeline HAMAP, which provides functional annotation for protein sequences to the same depth and quality as UniProtKB/Swiss-Prot, and the World Wide Web Consortium (W3C) standards Resource Description Framework (RDF) and SPARQL (a recursive acronym for the SPARQL Protocol and RDF Query Language). We translate complex HAMAP rules into the W3C standard SPARQL 1.1 syntax, and then apply them to protein sequences in RDF format using freely available SPARQL engines. This approach supports the generation of annotation that is identical to that generated by our own in-house pipeline, using standard, off-the-shelf solutions, and is applicable to any genome or proteome annotation pipeline. **Conclusions:** HAMAP SPARQL rules are freely available for download from the HAMAP FTP site, ftp://ftp.expasy.org/databases/hamap/sparql/, under the CC-BY-ND 4.0 license. The annotations generated by the rules are under the CC-BY 4.0 license. A tutorial and supplementary code to use HAMAP as SPARQL are available on GitHub at https://github.com/sib-swiss/HAMAP-SPARQL, and general documentation about HAMAP can be found on the HAMAP website at https://hamap.expasy.org.

Keywords: protein; function; prediction; SPARQL

Introduction

Continuing technological advances have reduced the costs of DNA sequencing enormously in recent years, leading to an explosion in the number of available whole-genome and metagenome sequences from all branches of the tree of life [1–5]. This wealth of sequence data presents exciting

Received: 28 June 2019; **Revised:** 30 November 2019; **Accepted:** 13 January 2020

opportunities for experimental and computational research into the evolution and functional capacities of individual organisms and the communities they form, but fully exploiting these data will require complete and accurate functional annotation of these genome and metagenome sequences. Resources for genome annotation such as RAST/MG-RAST [6, 7], IMG/M [8], the NCBI genome annotation pipeline [9], InterPro [10], TIGRFAMS [11], and HAMAP [12] exploit information from experimentally characterized sequences to infer functions for uncharacterized homologs. While the underlying principles of these resources are undoubtedly similar, a lack of shared annotation standards and a suitable shared technical framework for annotation hamper efforts to use and combine them.

In this work, we use the HAMAP system to demonstrate technical solutions that could facilitate the combination and reuse of functional genome annotation systems from any provider. HAMAP classifies and annotates protein sequences using a collection of expert-curated protein family signatures and annotation rules. Swiss-Prot curators build HAMAP rules as part of an integrated workflow that includes curation of experimentally characterized template entries in UniProtKB/Swiss-Prot, as well as curation of the associated rule and protein family signature (encoded as a generalized profile). HAMAP rules annotate family members to the same level of detail and quality as the expert-curated UniProtKB/Swiss-Prot records on which they are based, combining family membership and residue dependencies to ensure a high degree of specificity [12].

The current implementation of HAMAP uses a custom rule format and annotation engine that are not easy to integrate into external pipelines. The HAMAP-Scan web service [13] is a good alternative for small research projects, but large genome-sequencing projects cannot depend on external web services to process large amounts of data. Our goal here was to develop a generic HAMAP rule format and annotation engine that is easily portable by external HAMAP users, using standard technologies that developers of other genome annotation pipelines could also adopt. To achieve this we have developed a representation of HAMAP annotation rules using the W3C standard SPARQL 1.1 syntax. SPARQL [14] is a query language for RDF [15], a core Semantic Web technology from the W3C. Our implementation allows users to apply HAMAP rules in SPARQL syntax to annotate protein sequences expressed as RDF using off-the-shelf SPARQL engines—without any need for a custom pipeline. If other annotation system providers were to adopt the same approach, it would then be possible to share and combine the annotation rules from multiple systems, execute them with any SPARQL engine, and compare the results.

Methods

To use a generic SPARQL engine to execute rule-based protein sequence annotation, we need the following input data: (i) annotation rules in SPARQL syntax, (ii) protein sequence records in RDF syntax, and (iii) protein sequence/signature matches in RDF syntax, including alignment information for positional annotations.

To keep the examples given in the figures short, we provide all RDF namespace prefixes declarations in Fig. 1 and omit these from subsequent figures. We use the UniProt core ontology and other ontologies used by UniProt, such as the Feature Annotation Location Description Ontology (FALDO) [16], which is also used in the RDF of Ensembl [17] and Ensembl Genomes [18], to describe sequence positions, and the EDAM ontology [19] to describe sequence/signature matches.

```
prefix rdf:
 <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix rdfs:
 <http://www.w3.org/2000/01/rdf-schema#>
prefix up:
 <http://purl.uniprot.org/core/>
prefix taxon:
 <http://purl.uniprot.org/taxonomy/>
prefix ec:
 <http://purl.uniprot.org/enzyme/>
prefix GO:
 <http://purl.obolibrary.org/obo/GO_>
prefix ECO:
 <http://purl.obolibrary.org/obo/ECO_>
prefix faldo:
 <http://biohackathon.org/resource/faldo#>
prefix rule:
 <http://purl.uniprot.org/hamap-rule/>
prefix signature:
 <http://purl.uniprot.org/hamap/>
prefix edam:
 <http://edamontology.org/>
prefix example:
 <URL space of your protein sequences>
```

Figure 1: RDF namespace declarations for prefixes used in other figures.

```
CONSTRUCT {
  ?target up:classifiedWith GO:0004055 , GO:0005524 .
}
WHERE {
  ?target a up:Protein ;
          rdfs:seeAlso signature:MF_00005 ;
          up:sequence ?targetSequence ;
          up:organism ?organism .

  VALUES ?bacteriaORarchaea {taxon:2 taxon:2157}
  ?organism rdfs:subClassOf+ ?bacteriaORarchaea  .

  MINUS { ?targetSequence up:fragment [] }
}
```

Figure 2: Part of the HAMAP rule for signature MF_00005 as a SPARQL CONSTRUCT query.

HAMAP annotation rules in SPARQL syntax

A HAMAP annotation rule consists of 2 parts: (i) the annotations and (ii) a set of conditions that must be satisfied in order to apply those annotations. The rule annotations can be expressed either by a CONSTRUCT block that returns the annotations as RDF triples or by an INSERT block that inserts these triples directly into an RDF store, while the rule conditions can be expressed by the WHERE clause of a SPARQL query. Fig. 2 shows part of the HAMAP rule for the signature MF_00005 as a SPARQL query. The CONSTRUCT block generates 2 annotations consisting of RDF triples for 2 Gene Ontology (GO) terms, providing that all conditions defined in the WHERE clause are satisfied. The conditions here are that the target must be a complete protein sequence, of bacterial or archaeal origin, and a member of the HAMAP family MF_00005 (i.e., matching the corresponding family signature).

Fig. 3 shows how the CONSTRUCT block of Fig. 2 can be extended to generate metadata for provenance and evidence for each annotation that the rule generates. We attribute the

```
CONSTRUCT {
  # annotation statements
  ?target up:classifiedWith GO:0004055 , GO:0005524 .
  # metadata statements
  _:1 a               rdf:Statement ;
      rdf:subject     ?target ;
      rdf:predicate   up:classifiedWith ;
      rdf:object      GO:0004055 ;
      up:attribution  _:3 .

  _:2 a               rdf:Statement ;
      rdf:subject     ?target ;
      rdf:predicate   up:classifiedWith ;
      rdf:object      GO:0005524 ;
      up:attribution  _:3 .

  _:3 up:source        rule:MF_00005 ;
      up:evidence      ECO:256 .
}
```

Figure 3: SPARQL CONSTRUCT block of Fig. 2 extended with metadata expressed as RDF reification quads.

```
example:P1
  a up:Protein ;
  up:sequence example:P1-seq ;
  up:organism taxon:83333 .

example:P1-seq
  a up:Simple_Sequence ;
  rdf:value """MTKQKLILAYSGGLDTSVAIKWLSKDYDVVAL
CMDVGEGKDLSVIKEKALLVGAIESIVLDVKDEFANDFVLPALQYGA
HYEGAYPLISALSRPLIAEKLVEVAHAQGATAVAHGCTGKGNDQVRF
EVSVAALDPSLEVIAPVREWKWSREEEIAYAKENNVPIPINLNSPYS
IDMNLWGRSNECGVLENPWTEPPQDAYALTVAPEDAPDQAEEVIIGF
EAGVPVSINGTAYPLAKLITELNIIAGAHGVGRIDHVENRLVGIKSR
EVYECPGATVLLKAHAALETITLTKDVAHFKPILSKQYAETIYNGLF
HAPLTKGLKAFLTATQQDVTGEVRVKLYKGNATVTGRQSAVSLYDEK
LATYTKEDAFDHEAAKGFIKLHGLAISTHASVHRQEGVKK""" .
```

Figure 4: Example protein record in an RDF format suitable for HAMAP SPARQL rules.

annotations to the HAMAP rule (MF_00005) and describe the type of the evidence with a value from the Evidence Code Ontology (ECO) [20]. We link the attribution to the annotations via RDF reification quads, which is verbose but is understood by all RDF syntaxes and data stores.

The original HAMAP rule implementation has 2 features that we have not yet implemented in this work. The first is the ability to call sequence analysis methods such as SignalP [21] and TMHMM [22] for the annotation of signal and transmembrane regions, which is not implemented here because these methods may not be available to external users. The second is precedence relationships between HAMAP rules, which are complex and apply to relatively few rules.

Protein sequence records in RDF syntax

HAMAP SPARQL rules require protein sequence records in a simple RDF format. Fig. 4 shows an example protein record with the identifier "P1" (example:P1). The rules require an identifier for the sequence (example:P1-seq) and the organism as an NCBI taxonomy identifier (taxon:83333). The actual protein sequence is provided as an IUPAC amino acid encoded string (in the rdf:value predicate of example:P1-seq) for positional annotations.

Protein sequence/signature matches in RDF syntax

HAMAP SPARQL rules require sequence/signature match data in an RDF format.

Fig. 4 shows an RDF representation of the sequence/signature match of the example protein "P1" (Fig. 4) and the HAMAP signature MF_00005. The core information is a triple that states that the protein (example:P1) matches the signature (signature:MF_00005).

For positional annotations, the rule needs the start and end positions of the match region on the sequence, as well as the alignment between sequence and signature. We describe this information with the EDAM and FALDO ontologies and use the alignment format returned by the PfTools v3 [23] and InterProScan [10] software.

A HAMAP rule specifies the sequence positions of feature annotations—such as active sites or binding regions—with respect to 1 or more experimentally characterized "template" sequences in UniProtKB/Swiss-Prot. The rule engine therefore requires the alignment(s) of the template sequence(s) to the rule's signature as input, and uses these to determine the corresponding positions on the template(s) and target sequence. A HAMAP rule may additionally require that the matching discrete position or range on the target sequence correspond to a specified amino acid or sequence motif, e.g., to check that an active site has the expected amino acid. This functionality can be implemented either in standard SPARQL 1.1 using the REPLACE, STRLEN, and CONCAT functions (see Supplementary Fig. S1 for an example), or via a custom SPARQL function (an example Java function for RDF stores that extends the Apache Jena ARQ SPARQL engine is given in Supplementary Fig. S2). We distribute the template sequence/signature alignments that are required for rule application together with the rules on our FTP site [24].

Simplifying the output from HAMAP rules for other annotation pipelines

HAMAP rules provide functional annotation in the form of free text and using controlled vocabularies and ontologies developed by UniProt and others. These include GO [25], the Enzyme Classification of the IUBMB ("EC numbers") [26] represented by the ENZYME database [27], and the Rhea database of biochemical reactions [28] based on the ChEBI ontology [29]. Each HAMAP rule provides all annotation fields required in UniProtKB. For users requiring only a subset of these annotations—such as enzymatic reactions described using Rhea, or protein functions, processes, and cellular components described using GO—it is possible to translate only the desired annotation types into SPARQL queries. We can also modify the CONSTRUCT/INSERT block of the queries to return the results as simple protein-annotation associations (see Table 1). This tabular result format can easily be loaded into a relational database or JSON-based document store and requires no further investment in a Semantic Web technology stack.

Results

Validation

We have tested the approach of executing rule-based annotation with a generic SPARQL engine with the data from the HAMAP

Table 1: Simple protein-annotation associations of HAMAP rule MF_00005 for UniProtKB entry B1YJ35

Protein	Annotation
uniprot:B1YJ35	"GO:0004055"
uniprot:B1YJ35	"GO:0005524"
uniprot:B1YJ35	"GO:0006526"
uniprot:B1YJ35	"GO:0005737"
uniprot:B1YJ35	"ec:6.3.4.5"
uniprot:B1YJ35	"rhea:10932"

```
example:P1
  rdfs:seeAlso signature:MF_00005 ;
  up:sequence example:P1-seq .

example:AN_ALIGNMENT_OPERATION
  a edam:operation_0300 ;
  edam:has_input signature:MF_00005 .

example:AN_ALIGNMENT
  a edam:data_0869 ;
  edam:is_output_of example:AN_ALIGNMENT_OPERATION ;
  faldo:location example:AN_ALIGNMENT_REGION ;
  rdf:value "KQKLILAYSGGLDTSVAIKWL--SKDYDVV" .

example:AN_ALIGNMENT_REGION
  a faldo:Region ;
  faldo:begin example:AN_ALIGNMENT_REGION_BEGIN ;
  faldo:end example:AN_ALIGNMENT_REGION_END .

example:AN_ALIGNMENT_REGION_BEGIN
  a faldo:ExactPosition ;
  faldo:reference example:P1-seq ;
  faldo:position 1 .

example:AN_ALIGNMENT_REGION_END
  a faldo:ExactPosition ;
  faldo:reference example:P1-seq ;
  faldo:position 28 .
```

Figure 5: Example protein sequence/signature match in RDF syntax.

and UniProtKB/Swiss-Prot releases 2019_10. We translated the HAMAP rules into SPARQL CONSTRUCT queries and the protein sequences into the RDF format described in Fig. 4. We generated the RDF representation of the sequence/signature matches, as illustrated in Fig. 5, directly from a relational database containing the results of PfTools v3.1 scans of UniProtKB/Swiss-Prot versus HAMAP for our internal HAMAP release pipeline. Other groups could achieve the same result by scanning their protein sequences with PfTools v3.2 [30], which has a new output option for RDF format, or with InterProScan and converting the XML result files into RDF format with the XSLT stylesheet that we provide for this purpose [31].

We tested 2 different open-source SPARQL engines (Virtuoso RDF 7.2 and Apache Jena TDB2 3.13.1) to execute our rules and validated the generated annotations by comparing them to those obtained from our custom platform. This platform, implemented in Scala/Java, uses as input files protein entries in FASTA format and HAMAP rules in their custom text format to generate annotations in UniProtKB format (text, XML, or RDF). The RDF data generated by the different systems was loaded into separate named graphs of an RDF database for comparisons using SPARQL queries to search for annotations unique to any of the 3 runs (see example query in Fig. 6). The existing custom HAMAP annotation pipeline and each of the 2 SPARQL engines generated identical annotations, except for those that depend on external sequence analysis methods and the evaluation of HAMAP rule precedence, which we did not implement here as described in section HAMAP annotation rules in SPARQL syntax.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?protein ?goOrKeyword
WHERE {
  GRAPH <http://example.org/hamap_original/> {
    ?protein up:classifiedWith
      ?goOrKeyword
  }
  MINUS {
    GRAPH <http://example.org/output-new/> {
      {
        ?protein up:classifiedWith
          ?goOrKeyword
      } UNION {
        ?protein up:classifiedWith/rdfs:subClassOf
          ?goOrKeyword
      }
    }
  }
}
```

Figure 6: Example query for comparison of annotations generated by the different systems, taking into account whether a system inserts the full GO or UniProt keyword hierarchy or only leaf nodes.

On a laptop with 8 cores, it takes ∼4 minutes to scan a small *Escherichia coli* proteome with the HAMAP signatures using PfTools v3.2 or InterProScan software, and 1 minute to execute the HAMAP rules with Apache Jena TDB2 (see the instructions in the tutorial [32]). This shows that the sequence/signature scanning step is the bottleneck in our system. Both steps, scanning and rule execution, could be run in an embarrassingly parallel fashion. A further optimization for high-performance computing would be to avoid HTTP communication by running the SPARQL query reader and processor in the same process.

An additional small benefit of the SPARQL representation is that SPARQL queries can be serialized in RDF and loaded into a SPARQL engine. We set up a server with our rules [33] that allows us to perform quality assurance on our rules by running analytical queries across them and SPARQL endpoints of other life science resources.

Discussion

Protein function annotation pipelines based on SPARQL

Here we have developed a SPARQL representation of HAMAP annotation rules that allows other groups with basic knowledge of this widespread standard technology to incorporate HAMAP in their own genome and proteome annotation pipelines. SPARQL can express all features of complex HAMAP rules, including the logic required for positional annotations, while freely available SPARQL engines provide a means to execute HAMAP rules without recourse to specialized software. This work demonstrates the feasibility of adopting SPARQL as a means to integrate existing functional annotation pipelines for genome-sequencing projects. This applies not only to expert curated rules from

A)

```
base <http://example.org/rnacentral/>
prefix SO: <http://purl.obolibrary.org/obo/SO_>
prefix rfam: <http://rfam.xfam.org/family/>
<URS0000638944>
  a SO:0000356 ;
  rdf:value """CUAGACCGAAGCUGCCAAGGUGCGUGAUCC
CUCGGUGAUGCCUUGAGUGUUGCUUCGCCAAAAAACAACCACACG
GCCUAGCCGAAUUUCUCAUU""" ;
  rdfs:seeAlso rfam:RF00003 .
```

B)

```
base <http://example.org/rnacentral/>
prefix SO: <http://purl.obolibrary.org/obo/SO_>
prefix GO: <http://purl.obolibrary.org/obo/GO_>
prefix rfam:<http://rfam.xfam.org/family/>
CONSTRUCT {
  ?rna SO:associated_with GO:0005685
}
WHERE {
  ?rna a SO:0000356 ;
  rdfs:seeAlso rfam:RF00003 .
}
```

Figure 7: (A) Hypothetical triples to describe a sequence entry from RNAcentral.org that is a member of the Rfam RNA family RF00003 (U1 spliceosomal RNA family). (B) Hypothetical rule associating RF00003 to the GO term GO:0005685 (definition: "A ribonucleoprotein complex that contains small nuclear RNA U1").

HAMAP and other systems but also annotation rules generated by automated approaches such as deep learning [34, 35], which require a feature vector to be expressed as an RDF triple as shown by Linked Open Data for Machine Learning (LOD4ML) [36]. SPARQL can also be adopted by those without access to specialized RDF triple stores by using a SPARQL to SQL mapping (such as that provided by any of the R2RML tools [37]) to execute SPARQL rules directly against data stored in a relational database. The main weakness of SPARQL is that, like many generic query engines, it tends to be computationally more expensive than a custom solution, but we have seen significant progress in the optimization of SPARQL engines in recent years [38].

An approach that is extensible to any domain of biology

While we have limited our demonstration to the use of SPARQL queries to formalize and execute protein annotation rules from HAMAP, there is nothing that ties the SPARQL approach to a particular domain of biology. Complete genome annotation requires identification and functional annotation of RNAs as well as proteins, and Fig. 7 provides a demonstration of how that annotation could be provided by SPARQL. Here a hypothetical SPARQL rule specifies functional (GO) annotation for an RNA sequence of RNAcentral [39] that is a member of the U1 spliceosomal RNA family as defined by Rfam [40].

The development of annotation rules for a given domain across different groups will require community standards for the representation of the relevant domain-specific annotation types. In this work we have used the RDF vocabularies of UniProt, which allowed us to easily compare the results of the SPARQL approach to those of our existing HAMAP rule annotation pipeline. As other appropriate community ontologies become available, our queries and SPARQL rules can be easily adapted.

Further work

We plan to further extend our implementation of HAMAP rules using SPARQL to include external method calls and deal with rule precedence (see Section HAMAP annotation rules in SPARQL syntax), and also develop a SPARQL representation for PROSITE, which provides protein domain annotation via a custom pipeline, ScanProsite [41]. HAMAP and PROSITE are 2 of the main components of the UniRule system of UniProt, which provides automatic annotation for unreviewed entries of UniProtKB/TrEMBL [42], and the approach described here could be extended to the entire UniRule system. The UniProt data model was recently extended to allow enzyme annotation using biochemical reaction data from the Rhea database [43], which will further extend the scope of HAMAP SPARQL rules to more specialized applications—such as the creation and annotation of draft networks of metabolic reactions [44, 45].

Availability of Supporting Source Code and Requirements

- Project name: HAMAP
- Project home page: https://hamap.expasy.org
- Other requirements: SPARQL 1.1–compliant RDF store, sequence/signature scanning software (e.g., PfTools v3.2, biotools:pfsearch, or InterProScan, RRID:SCR_005829)
- License: CC-BY-ND 4.0
- RRID:SCR_007701

Availability of Supporting Data and Materials

The data sets supporting the results of this article are available in the GigaDB repository [46].

Additional Files

Supplementary Information S1. Map position on template to target using SPARQL 1.1. standard functions.
Supplementary Information S2. Java Apache Jena ARQ Custom Function.
Supplementary Information S3. XSLT to covert InterProScan XML output to minimal RDF for HAMAP.

Abbreviations

ECO: Evidence Code Ontology; FALDO: Feature Annotation Location Description Ontology; GO: Gene Ontology; HAMAP: High-quality Automated and Manual Annotation of Proteins; IUPAC: International Union of Pure and Applied Chemistry; JSON: JavaScript Object Notation; LOD4ML: Linked Open Data for Machine Learning; NCBI: National Center for Biotechnology Information; RDF: Resource Description Framework; SIB: Swiss Institute of Bioinformatics; SPARQL: SPARQL Protocol and RDF Query Language; W3C: World Wide Web Consortium.

Competing interests

The authors declare that they have no competing interests.

Funding

HAMAP activities at the SIB are supported by the Swiss Federal Government through the State Secretariat for Education, Research and Innovation SERI, and the Swiss National Science

Foundation (SNSF). The development of the HAMAP SPARQL rules was also supported by the ELIXIR Implementation study on “A microbial metabolism resource for Systems Biology.” Funding for open access charge: SERI.

Authors’ contributions

J.B. designed the HAMAP-SPARQL system, implemented the software to convert HAMAP rules to SPARQL, ran the analysis, and co-wrote the manuscript. E.d.C. implemented the original HAMAP pipeline and assisted with the translation of the rules to SPARQL. D.B. implemented software to extract sequence/signature matches in RDF format from an Oracle database. S.G. helped to optimize SPARQL queries and define the rule data model. B.A.C. helped to define the RDF format for sequence/signature matches. A.H.A., E.C., C.H., P.M., I.P., and C.R. curate Swiss-Prot entries and HAMAP rules used in this manuscript. I.X. participated in the planning of the project. N.R. tested and revised the tutorial and co-wrote the manuscript. A.B. co-wrote and edited the manuscript. All authors provided critical feedback on the project.

Acknowledgements

We thank Dr. Marco Pagni of the SIB Swiss Institute of Bioinformatics for interesting discussions and critical reading of the manuscript.

References

1. Lewin HA, Robinson GE, Kress WJ, et al. Earth BioGenome Project: sequencing life for the future of life. Proc Natl Acad Sci U S A 2018;**115**(17):4325–33.
2. Mukherjee S, Seshadri R, Varghese NJ, et al. 1,003 reference genomes of bacterial and archaeal isolates expand coverage of the tree of life. Nat Biotechnol 2017;**35**(7):676–83.
3. Paez-Espino D, Eloe-Fadrosh EA, Pavlopoulos GA, et al. Uncovering Earth’s virome. Nature 2016;**536**(7617):425–30.
4. Thompson LR, Sanders JG, McDonald D, et al. A communal catalogue reveals Earth’s multiscale microbial diversity. Nature 2017;**551**(7681):457–63.
5. Tighe S, Afshinnekoo E, Rock TM, et al. Genomic methods and microbiological technologies for profiling novel and extreme environments for the Extreme Microbiome Project (XMP). J Biomol Tech 2017;**28**(1):31–9.
6. Meyer F, Bagchi S, Chaterji S, et al. MG-RAST version 4-lessons learned from a decade of low-budget ultra-high-throughput metagenome analysis. Brief Bioinform 2017.
7. Overbeek R, Olson R, Pusch GD, et al. The SEED and the Rapid Annotation of microbial genomes using Subsystems Technology (RAST). Nucleic Acids Res 2014;**42**(Database issue):D206–14.
8. Chen IA, Markowitz VM, Chu K, et al. IMG/M: integrated genome and metagenome comparative data analysis system. Nucleic Acids Res 2017;**45**(D1):D507–16.
9. Haft DH, Dicuccio M, Badretdin A, et al. RefSeq: an update on prokaryotic genome annotation and curation. Nucleic Acids Res 2018;**46**(D1):D851–60.
10. Mitchell AL, Attwood TK, Babbitt PC, et al. InterPro in 2019: improving coverage, classification and access to protein sequence annotations. Nucleic Acids Res 2019;**47**(D1):D351–60.
11. Haft DH, Selengut JD, Richter RA, et al. TIGRFAMs and genome properties in 2013. Nucleic Acids Res 2013;**41**(Database issue):D387–95.
12. Pedruzzi I, Rivoire C, Auchincloss AH, et al. HAMAP in 2015: updates to the protein family classification and annotation system. Nucleic Acids Res 2015;**43**(Database issue):D1064–70.
13. HAMAP-Scan web service. https://hamap.expasy.org/hamap_scan.html. Accessed 30 November 2019.
14. SPARQL Query Language for RDF. 2013. https://www.w3.org/2001/sw/wiki/SPARQL. Accessed 30 November 2019.
15. Resource Description Framework (RDF). 2014. https://www.w3.org/RDF. Accessed 30 November 2019.
16. Bolleman JT, Mungall CJ, Strozzi F, et al. FALDO: a semantic standard for describing the location of nucleotide and protein feature annotation. J Biomed Semantics 2016;**7**:39.
17. Zerbino DR, Achuthan P, Akanni W, et al. Ensembl 2018. Nucleic Acids Res 2018;**46**(D1):D754–61.
18. Kersey PJ, Allen JE, Allot A, et al. Ensembl Genomes 2018: an integrated omics infrastructure for non-vertebrate species. Nucleic Acids Res 2018;**46**(D1):D802–8.
19. Ison J, Kalas M, Jonassen I, et al. EDAM: an ontology of bioinformatics operations, types of data and identifiers, topics and formats. Bioinformatics 2013;**29**(10):1325–32.
20. Chibucos MC, Mungall CJ, Balakrishnan R, et al. Standardized description of scientific evidence using the Evidence Ontology (ECO). Database (Oxford) 2014;**2014**, doi:10.1093/database/bau075.
21. Petersen TN, Brunak S, von Heijne G, et al. SignalP 4.0: discriminating signal peptides from transmembrane regions. Nat Methods 2011;**8**(10):785–6.
22. Sonnhammer EL, von Heijne G, Krogh A. A hidden Markov model for predicting transmembrane helices in protein sequences. Proc Int Conf Intell Syst Mol Biol 1998;**6**:175–82.
23. Schuepbach T, Pagni M, Bridge A, et al. pfsearchV3: a code acceleration and heuristic to search PROSITE profiles. Bioinformatics 2013;**29**(9):1215–7.
24. HAMAP rules in SPARQL syntax. 2019. ftp://ftp.expasy.org/databases/hamap/sparql/. Accessed 5 December 2019.
25. The Gene Ontology Consortium. The Gene Ontology Resource: 20 years and still GOing strong. Nucleic Acids Res 2019;**47**(D1):D330–8.
26. McDonald AG, Boyce S, Tipton KF. ExplorEnz: the primary source of the IUBMB enzyme list. Nucleic Acids Res 2009;**37**(Database issue):D593–7.
27. Bairoch A. The ENZYME database in 2000. Nucleic Acids Res 2000;**28**(1):304–5.
28. Lombardot T, Morgat A, Axelsen KB, et al. Updates in Rhea: SPARQLing biochemical reaction data. Nucleic Acids Res 2018;**47**(D1):D596–600.
29. Hastings J, Owen G, Dekker A, et al. ChEBI in 2016: improved services and an expanding collection of metabolites. Nucleic Acids Res 2016;**44**(D1):D1214–9.
30. PfTools v3. 2013. https://github.com/sib-swiss/pftools3. Accessed 30 November 2019.
31. XSLT stylesheet to convert InterProScan XML to RDF. 2019. https://github.com/sib-swiss/HAMAP-SPARQL/blob/master/src/main/xlst/interproToRdf.xslt. Accessed 30 November 2019.
32. HAMAP as SPARQL tutorial. 2019. https://github.com/sib-swiss/HAMAP-SPARQL. Accessed 30 November 2019.
33. HAMAP SPARQL server. 2019. https://hamap.expasy.org/sparql. Accessed 30 November 2019.
34. Fa R, Cozzetto D, Wan C, et al. Predicting human protein function with multi-task deep neural networks. PLoS One 2018;**13**(6):e0198216.

35. Kulmanov M, Khan MA, Hoehndorf R, et al. DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. Bioinformatics 2018;**34**(4):660–8.
36. LOD4ML: Linked Open Data for Machine Learning. 2017. http://lod4ml.org. Accessed 30 November 2019.
37. R2RML: RDB to RDF Mapping Language. 2012. https://www.w3.org/TR/r2rml/. Accessed 30 November 2019.
38. Schmidt M, Meier M, Lausen G. Foundations of SPARQL query optimization. In: Proceedings of the 13th International Conference on Database Theory. ACM;2010:4–33.
39. The RNAcentral Consortium. RNAcentral: a comprehensive database of non-coding RNA sequences. Nucleic Acids Res 2017;**45**(D1):D128–34.
40. Kalvari I, Argasinska J, Quinones-Olvera N, et al. Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families. Nucleic Acids Res 2018;**46**(D1):D335–42.
41. Sigrist CJ, de Castro E, Cerutti L, et al. New and continuing developments at PROSITE. Nucleic Acids Res 2013;**41**(Database issue):D344–7.
42. The UniProt Consortium. UniProt: a worldwide hub of protein knowledge. Nucleic Acids Res 2019;**47**(D1):D506–15.
43. Morgat A, Lombardot T, Coudert E, et al. Enzyme annotation in UniProtKB using Rhea. Bioinformatics 2019; btz817.
44. Faria JP, Rocha M, Rocha I, et al. Methods for automated genome-scale metabolic model reconstruction. Biochem Soc Trans 2018;**46**(4):931–6.
45. Moretti S, Martin O, Van Du Tran T, et al. MetaNetX/MNXref–reconciliation of metabolites and biochemical reactions to bring together genome-scale metabolic networks. Nucleic Acids Res 2016;**44**(D1):D523–6.
46. Bolleman JT, de Castro E, Baratin D, et al. Supporting data for "HAMAP as SPARQL rules—A portable annotation pipeline for genomes and proteomes." GigaScience Database. 2020. http://dx.doi.org/10.5524/100683.