Semantic Beacons: a framework to support federated querying over genomic variants and public Knowledge Graphs

Alexandrina Bodrug-Schepers[1,†], Hugo Chabane[2,†], Gabriela Montoya[2], Patricia Serrano-Alvarado[2], Richard Redon[1] and Alban Gaignard[1,3]

[1]*Nantes Université, CNRS, INSERM, l'institut du thorax, F-44000 Nantes, France*

[2]*Nantes Université, LS2N, Nantes, France*

[3]*IFB-core, Institut Français de Bioinformatique (IFB), CNRS, INSERM, INRAE, CEA, 91057 Evry, France*

Abstract

Comprehensive genomic data exchange is challenging but critical for future research. Beacon API networks, promoted by initiatives like GA4GH, facilitate genomic variation data discovery while preserving privacy and data ownership. However, their use is often limited by the need for costly storage, compute intensive data pre-processing, and periodic updates as genomic knowledge constantly progresses. This work proposes an on-the-fly approach for enriching genomic variants with biological annotations provided by established knowledge bases. It thus reduces the computational load and processing time. We explore integrating open and interoperable life sciences knowledge graphs with sensitive health genomic data discoverable through Beacon APIs. We propose this federated framework as a step towards increasing FAIRness of genomic data.

Keywords

Knowledge Graphs, Ontologies, Federated queries, RML semantic mappings, Beacon API, FAIR

1. Introduction

Understanding rare or multi-factorial diseases requires the setup of large cohorts as well as collaborative interdisciplinary research covering physiopathology, medical image processing and genomic sequencing in patients. These collaborative efforts require the combination of multi-source data, each with distinct access methods and security requirements. For instance, medical imaging data is increasingly recognized as sensitive, while genomic data is always required to be securely stored on-site in dedicated infrastructures. However, genomic variation retrieved from patients are often combined with annotations originating from public knowledge bases to evaluate their potential consequences to an already known disease or disorder. In addition, researchers generally need to reproduce their findings on other cohorts and need to identify relevant genomic datasets to reuse. Consequently, sharing this data is essential to accelerate research on multi-factorial or rare diseases, as each organisation hosts valuable sensitive genomic health data. For now, due to regulatory and security constraints, it is very challenging to develop federated systems allowing queries over human health data, including genomic variation and other types of biomedical data.

The difficulties of sharing sensitive genomic health data is partly addressed with the development of the Beacon API network [1], promoted by international initiatives such as Global Alliance for Genomics and

[†]These authors contributed equally.

alexandrina.bodrug@univ-nantes.fr (A. Bodrug-Schepers); alban.gaignard@univ-nantes.fr (A. Gaignard)

Health (GA4GH)[1]. These APIs allow the discovery of genomic variations while not disclosing sensitive fine-grained information, protecting patient privacy and data ownership [2]. Prior to making data discoverable through a Beacon API, some pre-processing is required [3] which is relatively costly, both in terms of storage space and computation time. To annotate genomic variants – and there can be millions for a single biological sample – several reference databases must be downloaded. These databases allow to describe, for instance, the function of a gene, the pathogenicity of a mutation, its frequency in a reference population, etc. All these annotations are critical for the biological interpretation of genomic variants. This presents several drawbacks: *i)* significant network bandwidth and storage space is required, *ii)* in cohorts with many biological samples to process, a high performance computing infrastructure is needed, and finally *iii)* when the state of knowledge evolves over time, it becomes necessary to rerun the entire workflow for updating the annotations. The solution we propose is to execute genomic variation annotation on-the-fly, leveraging up-to-date, open and trusted reference knowledge bases such as Uniprot [4], Genome Aggregation Database (gnomAD) [5] or Ensembl [6]. This would save more time for analysis and cut back data processing steps.

The Findable, Accessible, Interoperable and Reusable (FAIR) principles and the development of public and interoperable Life Science knowledge graphs make it possible for data scientists to query, reason and integrate massive, diverse and decentralized knowledge bases [7, 8, 9]. For instance, in the context of rare diseases, with a very limited amount of data describing affected patients, data scientists can leverage the Human Phenotype Ontology (HPO) [10] and reason on phenotype hierarchies for query expansion. Similarly, data scientists can benefit from the Gene Ontology (GO) [11] in the Uniprot knowledge base to identify target proteins associated to a typical biological function.

In this work, we address the following research question: **Is it possible to integrate on-the-fly public knowledge graphs with state-of-the art genomic data discovery APIs ?**

The main contributions of this work are *i)* a system architecture for on-the-fly FAIRification of genomic variation data, *ii)* a semantic mapping for aligning Beacon genomic variations to reference life science ontologies, and *iii)* a concrete federated query showcasing the integration of local genomic variation and public knowledge graphs.

The remainder of this paper is organized as follows. Section 2 introduces the biological needs for querying both public KGs and genomic variants. Section 3 describes the synthetic genomic data, the domain specific ontologies and the strategy for the on-the-fly data FAIRification. Section 4 presents the technical solution to access Beacon APIs as semantic datasources, and the federated query responding to our motivation use case.

2. Motivating use case

We focus on Intracranial aneurysms (ICA), a neurovascular disease characterized by saccular deformations of brain vessels. ICA rupture, although rare, often results in death or severe disabilities. The causes and mechanisms of ICA formation, stabilization or progression towards rupture remain unknown [12, 13, 14, 15]. Information about the genomic variation found in patients suffering from ICA is essential to accelerate research on this topic, as ICA formation is known to be at least partially due to genomic factors.

Figure 1 introduces our motivating scenario. Imagine that Alice is a data scientist studying blood vessel walls and their adaptation to varying blood flows. For her research project Alice has access to the genomic variants of patients carrying brain vessel deformities i.e., aneurysms. She is interested in the subset of genomic variants found in genes coding for proteins involved in blood vessel formation i.e.,

[1] https://www.ga4gh.org

angiogenesis. Here is a typical question she wants to answer: *"Which genomic variants reside in genes involved in angiogenesis among patients with intracranial aneurysms?"*.

Answering Alice's question requires the combination of public knowledge provided by databases such as UniprotKB (protein and their functional annotation) and Wikidata (protein and their corresponding gene location) with genomic variants found in Alice's experimental data obtained from patients.

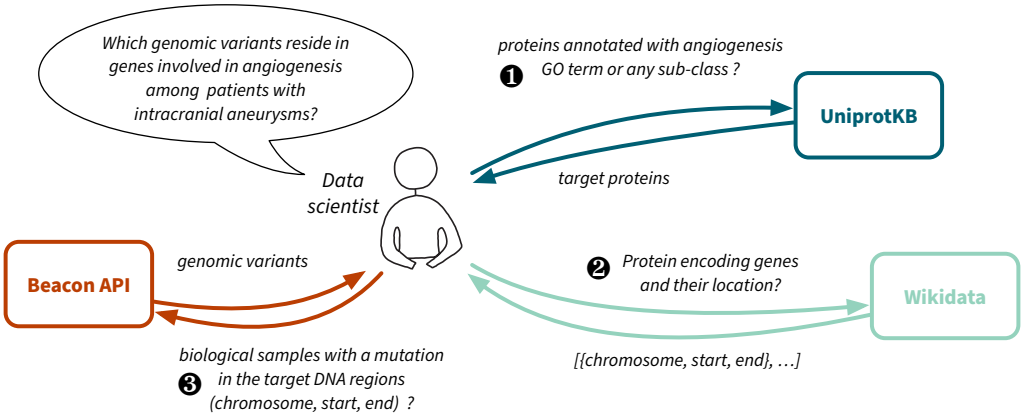

Figure 1: Integration of public databases with experimental data. Beacon hosts Alice's experimental data and metadata about genomic variation, UniprotKB is a vast resource describing proteins, including Gene Ontology describing molecular function, cellular components and biological processes, and Wikidata provides gene locations.

In step ❶, it is necessary to retrieve a list of proteins associated with "angiogenesis" using the Gene Ontology term *GO:0001525*. Since UniprotKB is a knowledge graph that hosts the Gene Ontology, it is possible to exploit its class hierarchy to automatically retrieve more specific concepts still relevant for Alice's research topic, for example "angiogenesis involved in coronary vascular morphogenesis" which corresponds to the term *GO:0060978*.

In step ❷, using Wikidata, it is possible to locate the genes coding for these proteins, using protein identifiers obtained from UniprotKB. Wikidata is instrumental in bridging the gap between angiogenesis associated proteins retrieved from UniprotKB and experimentally obtained genomic variants found in the Beacon API ❸, since the variants may be located within genes.

Succinctly, a genomic variant location can be described in four parts: the reference genome assembly it was found in, the chromosome, the start and the end coordinates. The variation part denotes a difference between the reference genome (called reference base) and the patient's genome (called alternate base).

Finally in step ❸, it is necessary to filter the millions of genomic variants found in aneurysm patients based on whether or not they are found within genes coding for proteins involved in angiogenesis.

In this work, we aim at alleviating the burden of manually integrating the heterogeneous data sources by proposing an automated approach based on federated queries. Thus, our goal is to propose a system architecture allowing to automatically execute the work plan of Figure 1 through a single query.

3. Approach

In this section we introduce our approach which consists in using relevant ontologies and public data sources (Section 3.1) and a mediator (Section 3.2) to semantically access a Beacon API currently containing synthetic genomic data.

3.1. Ontologies and Knowledge Graphs for synthetic genomic variations

Synthetic genomic dataset We used an in house simulated data to mimic experimental data. The simulation was done using genomic variants from the 1000 genomes project [2]. We extracted genomic variants contained in a subset of genes involved in angiogenesis. The human genome reference assembly was GRCh38.

Semantic representation of genome variants Sequence Ontology (SO), FALDO [16] and Genotype Ontology were used to describe the genomic variants. Sequence Ontology is used to describe the features and attributes of biological sequences. FALDO aims at uniformly representing precise locations within genomic sequences. A genomic variant is defined as a difference between nucleotide base(s) in an individual and the reference genome assembly of choice at a known location. We used the term *SO:0001059* (sequence alteration) to describe a genomic variant. We used FALDO to connect the *SO:0001059* concept to its genomic location. We followed the FALDO data schema to represent the location on the reference genome (chromosome, start, end). To designate the chromosome sequence we used the term *SO:0000353* described as *a sequence of nucleotides that has been algorithmically derived from an alignment of two or more different sequences*. We used the Genotype Ontology to represent the reference nucleotide(s) found in the reference genome assembly and the alternate nucleotide(s) found in patients. Figure 2 summarizes how to semantically represent and share genomic variants discoverable through Beacon APIs.

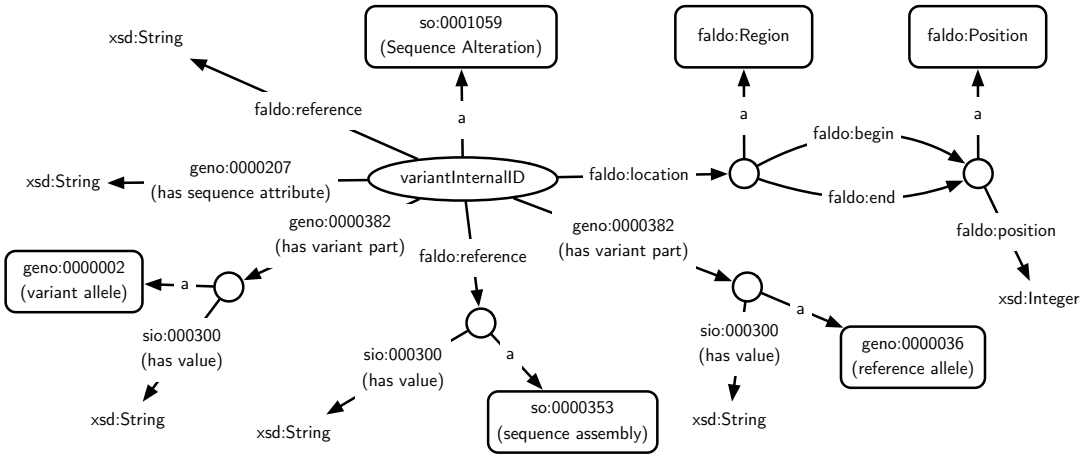

Figure 2: Semantic schema leveraging SO, FALDO and GENO ontologies for representing genomic variants. Circles are instances, rectangles are classes and edges are properties.

Protein functional annotation Gene Ontology (GO) is a biomedical ontology [17] used in many life science data or knowledge bases. It hierarchically represents molecular functions, biological processes and cellular components. For instance, "angiogenesis", a biological process identified with *GO:0001525*, is precisely defined as follows: *"Blood vessel formation when new vessels emerge from the proliferation of pre-existing blood vessels."*. This GO term is used in the monarch initiative database [18] to annotate 2,687 genes, and in Uniprot [19] to annotate 75,339 proteins. We used GO as a source of annotation for proteins. UniprotKB is the Knowledge Base of the UNIversal PROTein resource that interconnects information from large and heterogeneous biological sources. It supports many ontologies such as Gene Ontology, Human Phenotype Ontology and (rare) disease ontologies. UniprotKB describes protein sequence, their functional annotation and their association with phenotypes, diseases or disorders. We used UniprotKB to retrieve lists of human proteins and the associated Gene Ontology terms. UniprotKB does not contain information about gene location as it focuses on protein sequences.

[2] https://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/1000G_2504_high_coverage/

Gene location Wikidata[3] is a general purpose knowledge graph. We used this resource to retrieve gene location information (reference genome assembly, chromosome, start and end coordinate) of lists of genes that encode for proteins previously retrieved from UniprotKB. Wikidata retrieves its gene location information from Ensembl, a long time established and trusted genome database.

3.2. On-the-fly FAIRification of genomic variations

Our motivations for integrating Beacon APIs as Linked Data rely in *i)* minimizing server-side costs and avoiding a local duplication of public databases when enriching experimental genomic data with biological annotations, *ii)* benefiting from "fresh" public data for biological annotations, externally maintained, and *iii)* increasing the FAIRness of genomic variation data by reusing state-of-the-art and community agreed ontologies.

Integrating non-RDF data into Linked Data is possible thanks to a plethora of solutions proposing semantic mappings. RML (RDF Mapping language) [20] [4] is a widely-used mapping language allowing to express rules that map heterogeneous data into the RDF data model. It is based on the W3C-standardized mapping language R2RML[5] proposed to transform relational databases into RDF datasets. RML has several advantages. It is source-agnostic as it supports several formats such as CSV, JSON and XML. This flexibility facilitates the integration of diverse data sources. It also allows to express transformation rules that can be reused and easily extended. Indeed, an RML mapping can be reused for several data sources sharing the same structure.

In the data integration process [21], once mappings are identified, they can be used to materialize the source data into a local graph (following the ETL process). They can also be used to virtualize the data source by creating a graph view of the legacy data, enabling dynamic query rewriting that transforms graph-based queries into equivalent queries for the underlying data source.

Our approach employs ODMTP (On-Demand Mapping using Triple Patterns) [22], a framework for integrating non-RDF datasets on-demand into Linked Data using the TPF (Triple Pattern Fragments) methodology [23]. TPF minimizes server-side costs by providing a lightweight triple-pattern interface, delegating query decomposition to the client. Initially, the TPF client receives a SPARQL query and decomposes it into multiple triple pattern queries(TPQs) which are processed one by one by the server.

ODMTP implements a TPF server. As shown in Figure 3, its specificity lies in converting triple pattern queries into HTTP requests compatible with the Beacon API. The JSON response from the API is then mapped to RDF triples using a predefined RML mapping. These triples are organized into fragments and returned to the client, along with metadata (such as the total triples count and pagination details) which enable the client to create an efficient execution plan. The client calculates the final result and ultimately sends it to the user.

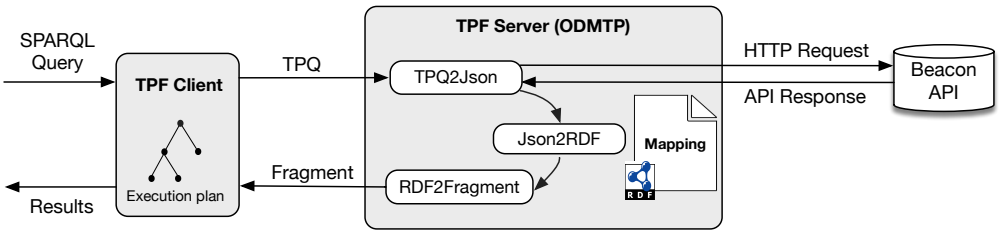

Figure 3: The ODMTP architecture, as described in [22], on top of a Beacon genomic variation dataset.

[3] https://www.wikidata.org/
[4] https://rml.io/specs/rml/
[5] https://www.w3.org/TR/r2rml/

4. Experimental results

4.1. Semantic mappings for Beacon API

Listing 1 shows an extract of the RML mappings for the Beacon API. In this listing we focus on the mapping that allows for obtaining the start position in the genes for the variations. Relevant prefixes are included in the Appendices.

Lines 1-15 allow for generating RDF triples that describe the start position of the genes. This information is obtained using an iterator based on the JSONPath expression given in line 5. The evaluation of this expression is a sequence and each element of that sequence represents a start position. We represent start positions as IRIs that are the concatenation of the `http://sembeacon.org/ressources/sb_` prefix, the constant *exactposition*, and the integer value of the position. Start positions are instances of the `faldo:ExactPosition` class (line 8). The integer value of the start position is available in the JSON file following the JSONPath expression given in line 12. For the JSON file given in Listing 2, this value is shown in line 8. These are the integer values that will be used in the federated query to relate the information in the Beacon API to the annotations available in Wikidata.

Existing tools such as Morph-KGC [24] allow for obtaining RDF triple using the RML mappings. For instance, using the mapping from Listing 1 on a JSON file (Listing 2), we obtain the RDF triples showed in Listing 3.

Listing 1: Extract from the RML mapping

```
1  _:BeginPositionMap a rr:TriplesMap ;
2     rml:logicalSource [
3        rml:source "reponse_beacon.json" ;
4        rml:referenceFormulation ql:JSONPath ;
5        rml:iterator "$.response.resultSets[*].results[*].variation.location.interval" ] ;
6     rr:subjectMap [
7        rr:template "http://ourlab.org/ressources/ol_exactposition{start.value}" ;
8        rr:class faldo:ExactPosition
9     ] ;
10    rr:predicateObjectMap [
11       rr:predicate faldo:position ;
12       rr:objectMap [ rml:reference "start.value" ;
13                      rr:termType rr:Literal ;
14                      rr:datatype xsd:integer ]
15    ] .
```

Listing 2: Partial Beacon's response

```
1  { "response": { "resultSets": [ { "results": [
2            { "variation": {
3                 "location": {
4                    "interval": {
5                       "start": {
6                          "value": 10093466 } } } } ] } ] }
                                  }
```

Listing 3: RDF representation (in Turtle) of the extract in Listing 2

```
1  ol:exactposition10093466 a faldo:ExactPosition ;
2        faldo:position 10093466 .
```

4.2. System architecture and implementation

To address our motivating use case, we designed and implemented a system based on federated SPARQL endpoints, on top of a FAIRified Beacon for genomic variation data. By FAIRified, we mean that it virtually exposes RDF data, reuses community agreed vocabularies (GENO, SO, etc.) and allows for semantic queries through the W3C standardized SPARQL language.

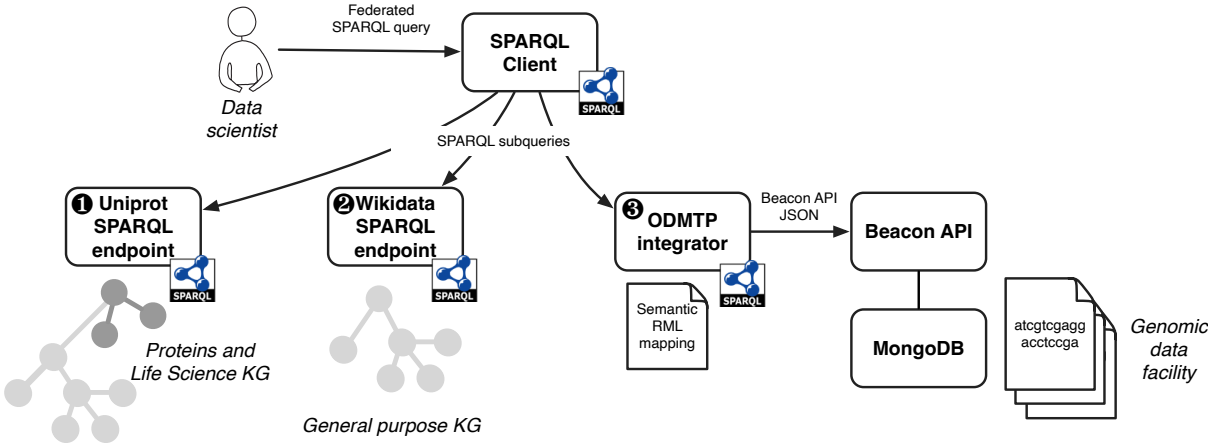

Figure 4: A decentralized architecture for combining biological annotations publicly accessible in reference a knowledge graphs, with genomic variations, accessible through Beacon APIs.

Figure 4 illustrates the deployed and reused technical components. The system takes first as input a federated SPARQL query that uses SERVICE clauses to specify the decentralized data source for each sub-query that is executed by an external service. The SPARQL client, implemented with Comunica [25], is responsible for the processing of the federated SPARQL queries. Parts of the query are sent to public knowledge graphs such as UniprotKB or Wikidata (❶, ❷), and the part of the query specific to the genomic variation data is sent to the ODMTP integrator (❸), responsible for the back-and-forth mapping between Beacon responses in JSON and RDF data following the semantic schema introduced in Figure 2. Finally, the reference implementation of the Beacon API is deployed on the same virtual machine as ODMTP to host our synthetic genomic dataset. More technical details are provided at our GitHub repository[6].

4.3. Combining public Life Science knowledge graphs and Beacon data

In this section we showcase a federated SPARQL query designed to answer our motivating use case. It uses three data sources, namely UniprotKB, Wikidata, and ODMTP for the local Beacon API. Listing 5 reports the full SPARQL query. Lines 2-7 describe a SERVICE clause with a basic graph pattern that is sent to the remote UniprotKB SPARQL endpoint. UniprotKB then provides a list of proteins, belonging to the human species (line 4), and classified with the Gene Ontology term "angiogenesis" and all its sub-classes. For that we rely on the *rdfs:subClassOf** SPARQL property path (line 6).

The second part of the query (lines 11-22) is sent to Wikidata. It is largely inspired by the query example provided by Uniprot[7]. Wikidata provides here the start position, the end position and the chromosome for locating the gene encoding for the given protein (UniprotKB).

Finally based on these locations, we filter the genomic variants provided by the local Beacon (lines 25-33). We use FALDO to refer to the location components (start, end, chromosome) and ensure that they are matching the locations provided by Wikidata (FILTER clause lines 30-33)

[6]https://github.com/SemanticBeacon/SemanticBeacon
[7]https://sparql.uniprot.org/.well-known/sparql-examples/?offset=15

Listing 4: Federated SPARQL query

```
SELECT * WHERE {
  SERVICE <https://sparql.uniprot.org/sparql> {
    ?protein a up:Protein ;
         up:organism taxon:9606 ;
         up:classifiedWith ?goTerm .
    ?goTerm rdfs:subClassOf* GO:0001525 .
  }
  BIND(SUBSTR(STR(?protein), STRLEN(STR(up:)) + 4) AS ?proteinID2) .

  SERVICE <https://query.wikidata.org/sparql> {
    ?wp wdt:P352 ?proteinID2 ;
         wdt:P702 ?wg .
    ?wg wdp:P644 ?wgss ;
         wdp:P645 ?wgse .
    ?wgss wdps:P644 ?startcoordinate ;
         wdpq:P1057/wdt:P1813 ?chromosome ;
         wdpq:P659/rdfs:label ?assembly .
    ?wgse wdps:P645 ?endcoordinate ;
         wdpq:P1057/wdt:P1813 ?chromosome ;
         wdpq:P659/rdfs:label ?assembly .
    FILTER(lang(?assembly) = "en")
    FILTER(STR(?assembly) = "genome assembly GRCh38")
  }

  ?variant a so:0001059 ;
    faldo:reference/sio:SIO_000300 ?chromosome ;
    faldo:location/faldo:begin/faldo:position ?v_start ;
    faldo:location/faldo:end/faldo:position ?v_end .

  FILTER( (((?v_start >= xsd:integer(?startcoordinate)) &&
            (?v_start <= xsd:integer(?endcoordinate)) ))
        || ((?v_end >= xsd:integer(?startcoordinate)) &&
            (?v_end <= xsd:integer(?endcoordinate))) )
}
LIMIT 10
```

5. Discussion and conclusion

In this paper, we propose a system leveraging semantic mappings to facilitate the federated querying of genomic variation data with public knowledge graphs.

One of the drawbacks of our prototype is the use of TPF to optimize and execute the queries. Following this approach, each API call retrieves a triple pattern fragment that often is considerably larger than what is specified in the query. This entails unnecessary data transfer and higher execution time. While still being aligned with genomics standards promoted by international consortia for genomic data sharing (Beacon), we extend their implementation so that they can be dynamically integrated through the FALDO, GENO, and GO community agreed life science ontologies. The major benefit of this approach is "data freshness", meaning that biological annotations are maintained externally and always up-to-date. However it can also be considered as a limitation since the reproducibility of genomics studies can be impacted by the reliability of external service, and the possibly unavailability of a given release of the public knowledge bases due to maintenance costs.

As future works, we aim at better covering the Beacon specification with semantic mappings. We plan to evaluate SPARQL federation engines [26, 27] in terms of query performance and source selection capability compared to native Beacon networks in the context of many participating data sources. In addition, to increase reproducibility and explainability of the federated query results, we aim at

enhancing the FAIRified genomic data with provenance metadata describing dataset versions as well as API request URLs and execution timestamps.

Acknowledgments

This work was partially funded by the French government, through the National Research Agency (ANR), under the "France 2030" program with reference ANR-22-PESN-0008. It was also partially funded by grants of the Labex Cominlabs excellence laboratory, managed by the French National Research Agency in the "Investing for the Future" program under reference ANR-10-LABX-07-01.

Declaration on Generative AI

During the preparation of this work, the author(s) used Chat-GPT-4 in order to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] Marc Fiume et al. "Federated discovery and sharing of genomic data using Beacons". en. In: *Nature Biotechnology* 37.3 (Mar. 2019). Publisher: Nature Publishing Group, pp. 220–224. ISSN: 1546-1696. DOI: 10.1038/s41587-019-0046-x. URL: https://www.nature.com/articles/s41587-019-0046-x (visited on 10/18/2024).

[2] Jordi Rambla et al. "Beacon v2 and Beacon networks: A "lingua franca" for federated data discovery in biomedical genomics, and beyond". In: *Human Mutation* 43 (2022), pp. 791–799. URL: https://api.semanticscholar.org/CorpusID:247497911.

[3] Manuel Rueda et al. "Beacon v2 Reference Implementation: a toolkit to enable federated sharing of genomic and phenotypic data". In: *Bioinformatics* 38.19 (Sept. 2022), pp. 4656–4657. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btac568. URL: https://doi.org/10.1093/bioinformatics/btac568 (visited on 03/20/2024).

[4] The UniProt Consortium. "UniProt: the Universal Protein Knowledgebase in 2023". In: *Nucleic Acids Research* 51.D1 (Jan. 2023), pp. D523–D531. ISSN: 0305-1048. DOI: 10.1093/nar/gkac1052. URL: https://doi.org/10.1093/nar/gkac1052 (visited on 10/29/2024).

[5] Monkol Lek et al. "Analysis of protein-coding genetic variation in 60,706 humans". en. In: *Nature* 536.7616 (Aug. 2016). Publisher: Nature Publishing Group, pp. 285–291. ISSN: 1476-4687. DOI: 10.1038/nature19057. URL: https://www.nature.com/articles/nature19057 (visited on 10/29/2024).

[6] Peter W Harrison et al. "Ensembl 2024". In: *Nucleic Acids Research* 52.D1 (Jan. 2024), pp. D891–D899. ISSN: 0305-1048. DOI: 10.1093/nar/gkad1049. URL: https://doi.org/10.1093/nar/gkad1049 (visited on 10/29/2024).

[7] Yasunori Yamamoto, Atsuko Yamaguchi, and Andrea Splendiani. "YummyData: providing high-quality open life science data". In: *Database: The Journal of Biological Databases and Curation* 2018 (2018). URL: https://api.semanticscholar.org/CorpusID:4026543.

[8] Maulik R. Kamdar and Mark A. Musen. "An empirical meta-analysis of the life sciences linked open data on the web". en. In: *Scientific Data* 8.1 (Jan. 2021). Publisher: Nature Publishing Group, p. 24. ISSN: 2052-4463. DOI: 10.1038/s41597-021-00797-y. URL: https://www.nature.com/articles/s41597-021-00797-y (visited on 10/17/2024).

[9] SIB Swiss Institute of Bioinformatics RDF Group Members. "The SIB Swiss Institute of Bioinformatics Semantic Web of data". In: *Nucleic Acids Research* 52.D1 (Oct. 2023), pp. D44–D51. ISSN: 0305-1048. DOI: 10.1093/nar/gkad902. eprint: https://academic.oup.com/nar/article-pdf/52/D1/D44/55040312/gkad902.pdf. URL: https://doi.org/10.1093/nar/gkad902.

[10] Sebastian Köhler et al. "The Human Phenotype Ontology in 2021". In: *Nucleic Acids Research* 49 (2020), pp. D1207–D1217. URL: https://api.semanticscholar.org/CorpusID:227259356.

[11] The Gene Ontology Consortium et al. "Gene Ontology: tool for the unification of biology". en. In: *Nature genetics* 25.1 (May 2000), p. 25. DOI: 10.1038/75556. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC3037419/ (visited on 10/29/2024).

[12] Romain Bourcier et al. "Understanding the Pathophysiology of Intracranial Aneurysm: The ICAN Project". In: *Neurosurgery* 80 (2017), pp. 621–626. URL: https://api.semanticscholar.org/CorpusID:3773283.

[13] Romain Bourcier et al. "Rare Coding Variants in ANGPTL6 Are Associated with Familial Forms of Intracranial Aneurysm." In: *American journal of human genetics* 102 1 (2018), pp. 133–141. URL: https://api.semanticscholar.org/CorpusID:34928519.

[14] Rafic Nader, Romain Bourcier, and Florent Autrusseau. "Using deep learning for an automatic detection and classification of the vascular bifurcations along the Circle of Willis". In: *Medical image analysis* 89 (2023), p. 102919. URL: https://api.semanticscholar.org/CorpusID:260834609.

[15] Olivia Rousseau et al. "Location of intracranial aneurysms is the main factor associated with rupture in the ICAN population". In: *Journal of Neurology, Neurosurgery, and Psychiatry* 92 (2020), pp. 122–128. URL: https://api.semanticscholar.org/CorpusID:225057631.

[16] Jerven T. Bolleman et al. "FALDO: a semantic standard for describing the location of nucleotide and protein feature annotation". In: *Journal of Biomedical Semantics* 7 (2014). URL: https://api.semanticscholar.org/CorpusID:196049489.

[17] Michael Ashburner et al. "Gene Ontology: tool for the unification of biology". In: *Nature Genetics* 25 (2000), pp. 25–29. URL: https://api.semanticscholar.org/CorpusID:10718909.

[18] Tim E. Putman et al. "The Monarch Initiative in 2024: an analytic platform integrating phenotypes, genes and diseases across species". In: *Nucleic Acids Research* 52 (2023), pp. D938–D949. URL: https://api.semanticscholar.org/CorpusID:265428484.

[19] Alex Bateman et al. "UniProt: the universal protein knowledgebase in 2021". In: *Nucleic Acids Research* 49 (2020), pp. D480–D489. URL: https://api.semanticscholar.org/CorpusID:227168406.

[20] Orme RML'E et al. "General anaesthesia using remifentanil for caesarean section in parturients with critical aortic stenosis: a series of four cases". In: *International Journal of Obstetric Anesthesia* 13.3 (2004), pp. 183–187.

[21] Aidan Hogan et al. "Knowledge graphs". In: *ACM Computing Surveys (Csur)* 54.4 (2021), pp. 1–37.

[22] Benjamin Moreau et al. "Querying non-RDF Datasets using Triple Patterns". In: *16th International Semantic Web Conference (ISWC2017) Poster&Demo session.* 2017.

[23] Ruben Verborgh et al. "Triple Pattern Fragments: a Low-cost Knowledge Graph Interface for the Web". In: *Journal of Web Semantics* 37–38 (Mar. 2016), pp. 184–206. ISSN: 1570-8268. DOI: doi:10.1016/j.websem.2016.03.003.

[24] Julián Arenas-Guerrero et al. "Morph-KGC: Scalable knowledge graph materialization with mapping partitions". In: *Semantic Web* 15.1 (2024), pp. 1–20. DOI: 10.3233/SW-223135. URL: https://doi.org/10.3233/SW-223135.

[25] Ruben Taelman et al. "Comunica: A Modular SPARQL Query Engine for the Web". In: *International Workshop on the Semantic Web*. 2018. URL: https://api.semanticscholar.org/CorpusID:52897313.

[26] Julien Aimonier-Davat et al. “FedUP: Querying Large-Scale Federations of SPARQL Endpoints”. In: *Proceedings of the ACM Web Conference 2024*. WWW ’24. Singapore, Singapore: Association for Computing Machinery, 2024, pp. 2315–2324. ISBN: 9798400701719. DOI: 10.1145/3589334.3645704. URL: https://doi.org/10.1145/3589334.3645704.

[27] Muhammad Saleem et al. “CostFed: Cost-Based Query Optimization for SPARQL Endpoint Federation”. In: *Procedia Computer Science* 137 (2018). Proceedings of the 14th International Conference on Semantic Systems 10th – 13th of September 2018 Vienna, Austria, pp. 163–174. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2018.09.016. URL: https://www.sciencedirect.com/science/article/pii/S1877050918316211.

Appendices

```
@prefix faldo: <http://biohackathon.org/resource/faldo#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix rml: <http://semweb.mmlab.be/ns/rml#> .
@prefix ql: <http://semweb.mmlab.be/ns/ql#> .
@prefix nv: <http://sembeacon.org/ressources/sb_> .
```

```
PREFIX up: <http://purl.uniprot.org/core/>
PREFIX taxon: <http://purl.uniprot.org/taxonomy/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX GO:<http://purl.obolibrary.org/obo/GO_>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX wdp: <http://www.wikidata.org/prop/>
PREFIX wdpq: <http://www.wikidata.org/prop/qualifier/>
PREFIX wdps: <http://www.wikidata.org/prop/statement/>

PREFIX geno: <http://example.org/geno/GENO_>
PREFIX sio: <http://semanticscience.org/resource/>
PREFIX so: <http://semanticscience.org/resource/SIO_>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

Listing 5: Federated SPARQL query

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX up: <http://purl.uniprot.org/core/>
PREFIX taxon: <http://purl.uniprot.org/taxonomy/>
PREFIX GO: <http://purl.obolibrary.org/obo/GO_>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX wdp: <http://www.wikidata.org/prop/>
PREFIX wdpq: <http://www.wikidata.org/prop/qualifier/>
PREFIX wdps: <http://www.wikidata.org/prop/statement/>


SELECT DISTINCT ?protein_name ?biotype ?gene_name ?assembly ?chromosome ?
    startcoordinate ?endcoordinate ?go_term ?wdgene
WHERE {
    # Get proteins annotated with GO:0001525 (angiogenesis) or its subclasses
    ?protein a up:Protein ;
             up:organism taxon:9606 ;
             rdfs:label ?protein_name ;
             up:encodedBy/skos:prefLabel ?gene_name ;
             up:classifiedWith ?go_term .

    ?go_term rdfs:subClassOf* GO:0001525 .
        ?go_term rdfs:label ?biotype .

  BIND(SUBSTR(STR(?protein), STRLEN(STR(up:)) + 4) AS ?wdprotein) .

  # Get genes coding for the proteins, and their coordinates in the hg38 assembly
  SERVICE <https://query.wikidata.org/sparql> {
    ?wp wdt:P352 ?wdprotein ;
        wdt:P702 ?wdgene .
        ?wdgene wdp:P644 ?wgss ;
        wdp:P645 ?wgse .
    ?wgss wdps:P644 ?startcoordinate ;
        wdpq:P1057/wdt:P1813 ?chromosome ;
        wdpq:P659/rdfs:label ?assembly .
    ?wgse wdps:P645 ?endcoordinate ;
        wdpq:P1057/wdt:P1813 ?chromosome ;
        wdpq:P659/rdfs:label ?assembly .
    FILTER(lang(?assembly) = "en")
    FILTER(STR(?assembly) = "genome assembly GRCh38")
  }
}
ORDER BY ?protein_name
```