CEN WORKSHOP AGREEMENT

CWA 18083

February 2024

ICS 07.080; 35.240.80

English version

Methodology for the construction of omics-related knowledge graphs from animal, vegetal and environmental data

This CEN Workshop Agreement has been drafted and approved by a Workshop of representatives of interested parties, the constitution of which is indicated in the foreword of this Workshop Agreement.

The formal process followed by the Workshop in the development of this Workshop Agreement has been endorsed by the National Members of CEN but neither the National Members of CEN nor the CEN-CENELEC Management Centre can be held accountable for the technical content of this CEN Workshop Agreement or possible conflicts with standards or legislation.

This CEN Workshop Agreement can in no way be held as being an official standard developed by CEN and its Members.

This CEN Workshop Agreement is publicly available as a reference document from the CEN Members National Standard Bodies.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and United Kingdom.

EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

Contents

European foreword

This CEN Workshop Agreement (CWA 18083:2024) has been developed in accordance with the CEN-CENELEC Guide 29 "CEN/CENELEC Workshop Agreements- A rapid standardization" and with the relevant provisions of CEN/CENELEC Internal Regulations-Part 2. It was approved by a Workshop of representatives of interested parties on 2024-01-31, the constitution of which was supported by CEN following the public call for participation made on 2023-11-23. However, this CEN Workshop Agreement does not necessarily include all relevant stakeholders.

The final text of CWA 18083:2024 was provided to CEN for publication on 2024-02-28.

Results incorporated in this CWA received funding from the European Union's HORIZON 2020 research and innovation programme under grant agreement number 952908.

The following organizations and individuals developed and approved this CEN Workshop Agreement:

- Eloy Hernandez (Chair) — Fundació Eurecat (EURECAT)
- Biotza Gutierrez (Vice-Chair) — Fundació Eurecat (EURECAT)
- Cristina Hernan (Secretariat) — UNE
- Aitor Corchero — NTT Data
- Andreia Salvador — Universidade Do Minho
- Armando Menéndez — Asociación de investigación de industrias cárnicas del Principado de Asturias (ASINCAR)
- Christian Espinoza — AkiNao
- Diana Duarte — Águas do Norte
- Enrique Gómez — Servicio regional de investigación y desarrollo agroalimentario (SERIDA)
- Isabel Gimeno — Servicio regional de investigación y desarrollo agroalimentario (SERIDA)
- María Fernández — Asociación española de criadores de ganado vacuno selecto de la raza Asturiana de los Valles (ASEAVA)
- María Alcina — Universidade Do Minho
- Nuria Canela — Fundació Eurecat (EURECAT)
- Patricia Cruz — Águas do Norte
- Xavier Domingo — Fundació Eurecat (EURECAT)

Attention is drawn to the possibility that some elements of this document may be subject to patent rights. CEN-CENELEC policy on patent rights is described in CEN-CENELEC Guide 8 "Guidelines for Implementation of the Common IPR Policy on Patent". CEN shall not be held responsible for identifying any or all such patent rights.

Although the Workshop parties have made every effort to ensure the reliability and accuracy of technical and non-technical descriptions, the Workshop is not able to guarantee, explicitly or implicitly, the correctness of this document. Anyone who applies this CEN Workshop Agreement shall be aware that neither the Workshop, nor CEN, can be held liable for damages or losses of any kind whatsoever. The use of this CEN Workshop Agreement does not relieve users of their responsibility for their own actions, and they apply this document at their own risk. The CEN Workshop Agreement should not be construed as legal advice authoritatively endorsed by CEN.

Introduction

The remarkable increase in analytical capabilities and the reduction of computational infrastructure costs have move on routine multi-omics studies to a new dimension. As a result, the data and information generated offer unprecedented new insights into intracellular mechanisms. In addition, they allow meaningful representations of signalling pathways and protein–protein interaction networks to be extracted, helping to understand these interrelated and complex biochemical processes.

In the development of the GLOMICAVE project, we have encountered the difficulty of organizing and analysing the omics data in a way that is computationally feasible. The aim is to be able to link the information atomically to make an understanding of the studies with each other. For this reason, a methodology is proposed for the generation and visualization of graphs that represented the relationships between genes, proteins, metabolites, and other relevant biological entities.

NOTE The GLOMICAVE project aims to develop a genotype-to-phenotype cloud platform based on Big Data and Artificial Intelligence (AI) analysis techniques, using large public and experimental omics data sets. Based on this information, the project has developed a set of tools for publicly available and experimental data, enhanced with automatic processing of scientific literature. Specific concepts of formats, databases, repositories, and other relevant technological elements that have been used for the construction of this methodology within the project are mentioned throughout the document.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement from CEN or this CEN Workshop.

1 Scope

This CWA (CEN Workshop Agreement) provides a methodology for the construction of knowledge graphs on topics related to non-human omics data, specifically in the domain related to animal, plant, and environmental studies in general. This methodology provides a guide for the representation and analysis of such data in these domains to facilitate the understanding and discovery of relevant relationships and patterns about the information contained in a potential knowledge graph.

This methodology is designed for application by researchers, scientists, and experts in the fields of genomics, proteomics, metabolomics, transcriptomics, and other omics fields related to non-human life in research work and projects in the field of semantics related to animals, plants, and the environment.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
application programming interface
API
set of rules and protocols that allow different software applications to communicate with each other

Note 1 to entry: An API defines the methods and data that an application can use to access and interact with the services or functionalities of another application, operating system, platform, or cloud service. This allows developers to use the capabilities of an application or service in their own applications without needing to know all the internal details of how that application works.

3.2
context
additional information that enriches the data, including metadata and relationships, to provide a deeper and more accurate understanding of the information and its interconnections

Note 1 to entry: Context helps to interpret data in a meaningful and specific way in various situations.

3.3
data model
visual representation of the data structure in a database or information system

Note 1 to entry: Data models are concepts used to help people know, understand, or simulate a subject that the model represents.

Note 2 to entry: Data models are a set of concepts that represent objects and their relationships in a particular application domain. It also determines the structure of the data represented in graphical form.

3.4
database
data storage system that enables the efficient retrieval, management, and manipulation of information

3.5
graph data base
system designed to store data in the form of interconnected nodes and relationships

Note 1 to entry: Graph database is especially useful for applications involving highly related data, such as social networks and recommender systems, allowing complex queries on interconnected data structures.

3.6
JavaScript Object Notation
JSON
lightweight data interchange format that uses a simple syntax to represent objects and structured data

Note 1 to entry: It is composed of key-value pairs and is commonly used in web applications to transmit information between servers and clients

3.7
JavaScript Object Notation for Linked Data
JSON-LD
JSON extension that allows the representation of structured and linked data on the web in a way that is readable by both humans and machines

Note 1 to entry: It uses a similar syntax to standard JSON but includes the ability to express semantic relationships and links to other resources on the web. JSON-LD is used to create linked data in web applications and facilitates interoperability and information exchange on the semantic web

3.8
knowledge network
data structure representing information in the form of nodes (also called vertices) and arcs (also called edges) connecting these nodes

Note 1 to entry: Each node represents an entity or concept, and the arcs represent relationships between these entities.

3.9
metadata
data that provides information about one or more aspects of the data

Note 1 to entry: Basically, it is data that describes other data in terms of type, structures, and syntax. The most common languages are RDF, XML and HTML.

3.10
ontology
explicit formal description of concepts in a concept domain and properties of each concept describing various characteristics and attributes of the concept and constraints

Note 1 to entry: An ontology together with a set of individual instances of classes constitutes a knowledge base.

3.11
plugins
additional software components that extend the functionality of a core application or system, allowing the incorporation of specific features

3.12
python

interpreted, object-oriented, high-level programming language with dynamic semantics

Note 1 to entry: Python has a simple, easy-to-learn syntax that emphasizes readability and therefore reduces the cost of program maintenance.

3.13
resource description framework

RDF

standard for data interchange that is used to represent highly interconnected data

Note 1 to entry: Each RDF statement is a three-part structure consisting of resources, where each resource is identified by a URI.

Note 2 to entry: Representing data in RDF allows information to be easily identified, disambiguated, and interconnected by artificial intelligence systems.

3.14
semantic model

explicit and formal specification of a shared conceptualization

[SOURCE: Thomas R. Gruber: A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199–220, 1993]

3.15
semantic relationship

specific and meaningful connection between concepts or entities in a particular context

Note 1 to entry: Differs from simple relationships by involving specific meanings and contexts, allowing for deep and specific understanding in data models.

3.16
SPARQL

RDF query language, or a semantic query language for databases, capable of retrieving and manipulating data stored in RDF format

3.17
triplestore

database management system designed to store and query data organised in triples (subject, predicate, object) according to the RDF standard

Note 1 to entry: Triplestore is essential for semantic web applications and for working with interconnected data.

4 Methodology for the construction of knowledge graphs

4.1 General

“Data repositories” are places designed to store, organize, and share data collected from a variety of sources. These repositories provide a centralized space to facilitate search, access, and distribution of data, and are essential for collaboration, analysis, and structured storage of information. These repositories typically store information in a structured or unstructured form and, such information is exposed to the public through APIs and/or data portals. These portals make the information stored in them available through a data model.

NOTE Data portals can be open (allow access to the information directly) or closed (allow access to the information through privileges granted by the platform administrator).

The data models used to expose the information are not standardized or normalized. Therefore, linking between one portal and another cannot be done directly, but requires a system or human intervention to link the information. In the absence of standardization in the display of studies, the information displayed usually encodes the studies and the underlying organisms differently, which increases the complexity of linking the information automatically.

To homogenize the information and, above all, to make explicit the underlying knowledge from the combination of multiple repositories, it is necessary to construct a knowledge graph housed within a semantic repository for further analysis.

Throughout this clause, the methodology of constructing graphs to facilitate the organization and analysis of omics data (animal, plant and environmental) is described, thereby improving the understanding of complex biological and ecological interactions between studies, and facilitating the understanding of study results and their interaction (Figure 1).

Although the methodology provides a sound guide for graph construction and analysis of omics data, its effectiveness may depend on the availability and quality of the data collected. For this reason, it is essential to ensure the accuracy and consistency of the data used to obtain reliable and meaningful results.

The proposed methodology encompasses several key stages for the construction of graphs in the context of non-human omics data. These stages include:

a) data collection and curation (see 4.2);

b) adoption of ontologies and taxonomies (see 4.3);

c) construction and maintenance of the knowledge graph (see 4.4);

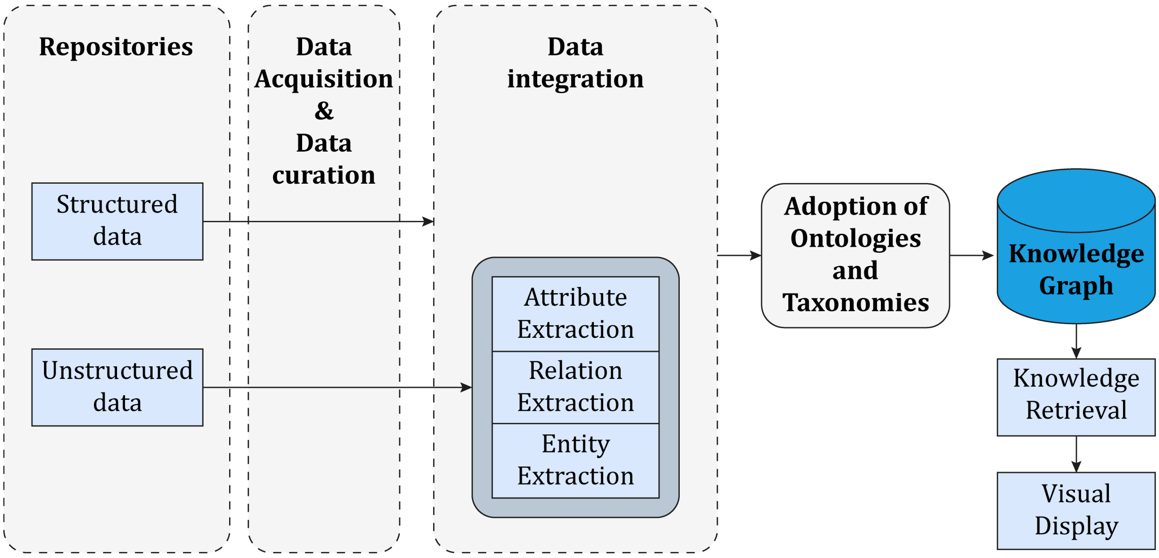


Figure 1 — Methodology for constructing knowledge graphs

4.2 Data collection and curation

4.2.1 General

Data collection and curation is the process by which relevant sources are identified to extract information for further analysis and exploration.

This phase of data collection and curation includes the following steps:

a) selection of the repositories of interest;

b) data collection;

c) data homogenization.

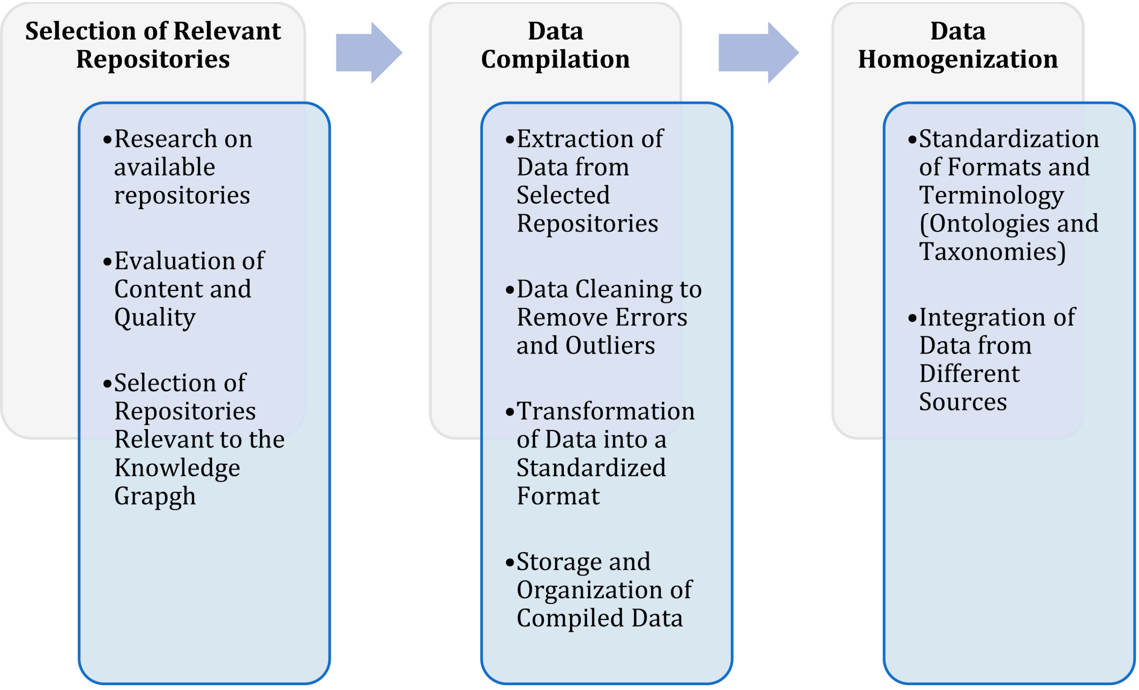

Selection of Relevant Repositories
•Research on available repositories
•Evaluation of Content and Quality
•Selection of Repositories Relevant to the Knowledge Grapgh
Data Compilation
•Extraction of Data from Selected Repositories
•Data Cleaning to Remove Errors and Outliers
•Transformation of Data into a Standardized Format
•Storage and Organization of Compiled Data
Data Homogenization
•Standardization of Formats and Terminology (Ontologies and Taxonomies)
•Integration of Data from Different Sources

Figure 2 — Data collection and curation phase

4.2.2 Selection of repositories of interest

For the selection of repositories, an evaluation of currently available omics repositories shall be carried out, which encompass a set of data related to various scientific and technological disciplines that focus on the study of biological molecules at a global level. These disciplines use advanced technologies to analyse and understand specific biological components or molecular patterns in whole biological systems. Some of the main types of omics data include genomics, transcriptomics, proteomics, metabolomics and phenomics, among others.

NOTE 1 The assessment indicates that the longer established omics, i.e. metabolomics, genomics, and transcriptomics, have more complete and well-established repositories, compared to the newer omics, where the repositories contain relatively few data sets.

The main goal of the various repositories is the preservation of raw data sets and to remove this burden of data preservation for scientific publishers. Importantly, most repositories support access to data through APIs that allow the development of software to automate this data acquisition process.

NOTE 2 While data licensing may be an issue with some specialized repositories, the major repositories do not appear to impose restrictions on data re-use.

For omics data, a repository shall be selected that meets the following requirements:

a) wide variety of data and relevance of data;

b) ease of access: It shall provide an application programming interface (API) that allows programmatic access to the data. This will facilitate automation of data retrieval through pre-developed plugins;

c) open access: It shall be freely available and openly accessible to the scientific community. This facilitates access and downloading of data for analysis;

d) data standardization: It shall use well-defined data formatting and metadata standards, which facilitates the extraction and processing of such data in a consistent manner;

e) scientific relevance: It shall host data generated by a wide range of scientific research and studies, which would make it a valuable source for addressing diverse research questions;

f) documentation and support: It shall provide detailed documentation and support for users, which will facilitate the understanding and use of its data.

NOTE 3 In the GLOMICAVE project we have selected the *Gene Expression Omnibus* (GEO) repository for genomic data and *Metabolights* for metabolomic data.

4.2.3 Data collection

To enable querying and converting data from different repository sources, and then integrating them into the knowledge graph, several plugins shall be created. These plugins shall access information using specific coded data exchange mechanisms to acquire data from each repository in an automated way. The objectives of these plugins shall be to:

a) to collect the data from the different selected data repositories (see 4.2.1);

b) to store the data in a structured database due to the large amount of information and in order not to saturate the source repositories with queries;

c) convert the relevance data into knowledge to be integrated into the knowledge graph.

The plugins shall be developed using the programming language that is most appropriate for the selected technology and that best fits the requirements of the different repositories.

NOTE 1 In the case of GLOMICAVE these plugins have been developed using the *Python* programming language, as it is one of the four programming languages (*Python, R, Julia* and *Scada*) supported by *Jupyter Notebook*. The latter is an interactive web-based environment for creating documents called “notebooks”. A notebook is a browser-based programming environment that takes user input and executes it, returning the results to the user. This notebook can contain code, text, mathematics, graphics, and rich media.

The data obtained through the plugins shall initially be stored in a structured database to maintain the provenance of the information and the initial state of the data. Subsequently, to build the knowledge graph, the ontology shall be instantiated with the retrieved data using the terms and relationships defined in the ontology (see 4.3). Finally, this information shall be stored in the knowledge graph database, which will allow querying this data and maintaining the knowledge graph through the semantic query language SPARQL.

NOTE 2 GLOMICAVE has used the so-called “S3 Buckets” within the AWS (Amazon Web Services) architecture as a cloud storage service, as it is particularly useful for storing large volumes of data efficiently.

4.2.4 Data homogenization

In the process of building the knowledge graph, it is essential to homogenize and standardize data to ensure consistency and interoperability in the knowledge graph. This process involves several essential steps:

a) data analysis: Conducting a thorough analysis of the available data to understand its structure, format and content;

b) identification of relevant variables: Identification of the key variables and attributes that are part of the data repositories and that are likely to be included in the ontology or taxonomy. These variables shall be clearly identified, defined and understandable to all users;

c) normalization and standardization: Normalize the data to ensure that it is in a consistent and standardized format. This involves converting units, dates and other values so that they are consistent throughout the ontology;

d) term mapping: Performing a mapping of terms and concepts to establish precise relationships between data. This involves defining equivalences, hierarchies and dependency relationships;

e) data integration: Integrating data from diverse sources ensuring that they are stored and represented in a consistent way in the ontology. This facilitates interoperability and effective querying;

f) documentation: Documenting the homogenization process in detail, including decisions made, transformations performed, and rules established. This documentation is crucial for understanding and maintaining the ontology in the future;

g) continuous updating and maintenance: Establishing methods and mechanisms to maintain the homogeneity of the data as new information is introduced and/or existing structures are modified. This involves periodic review and, if necessary, adjustments to the homogenization process.

4.3 Adoption of ontologies and taxonomies

4.3.1 General

The adoption of ontologies and taxonomies is the process within the methodology that allows structuring knowledge and defining the rules to transform implicit knowledge into explicit knowledge.

To address the construction of the semantic model, the use of existing ontology construction methodologies that facilitate the reuse of terms and definitions at both general (domain agnostic) and domain level is recommended.

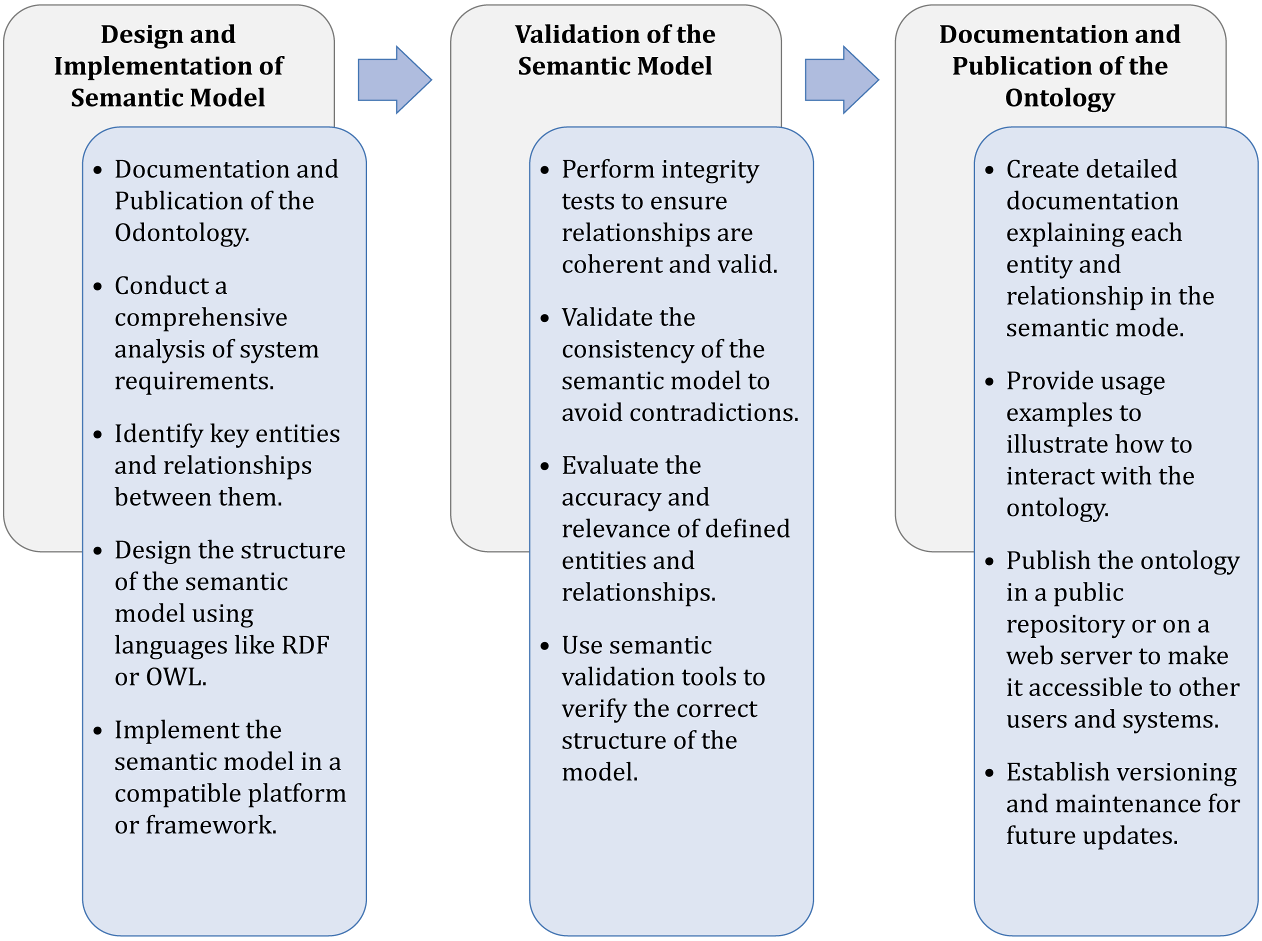
NOTE In GLOMICAVE the methodology used for the creation and maintenance of the semantic model is SAMOD [1].

The adoption of ontologies and taxonomies includes the following phases:

a) design and implementation of the semantic model;

b) validation of the semantic model;

c) documentation and publication of the ontology.



Design and Implementation of Semantic Model

- Documentation and Publication of the Odontology.
- Conduct a comprehensive analysis of system requirements.
- Identify key entities and relationships between them.
- Design the structure of the semantic model using languages like RDF or OWL.
- Implement the semantic model in a compatible platform or framework.

Validation of the Semantic Model

- Perform integrity tests to ensure relationships are coherent and valid.
- Validate the consistency of the semantic model to avoid contradictions.
- Evaluate the accuracy and relevance of defined entities and relationships.
- Use semantic validation tools to verify the correct structure of the model.

Documentation and Publication of the Ontology

- Create detailed documentation explaining each entity and relationship in the semantic mode.
- Provide usage examples to illustrate how to interact with the ontology.
- Publish the ontology in a public repository or on a web server to make it accessible to other users and systems.
- Establish versioning and maintenance for future updates.

Figure 3 — Adoption phase of ontologies and taxonomies

4.3.2 Design and implementation of the semantic model

The semantic model shall provide a uniform structure for describing experimental data sets obtained from the selected repositories (see 4.2.1). This metadata shall be automatically collected from the different repositories and converted into a standardized representation. The metadata shall facilitate automated analysis and interpretation of the experimental data sets.

This metadata should be used as a data index, allowing the user to identify and reuse relevant data sets for large-scale analyses more efficiently.

The design and implementation of the semantic model shall:

a) support data representation to enable homogenization between data sources and increase the knowledge base;

b) provide metadata and contextual information to interconnect scientific results, in this case related to multi-omics repositories;

c) generate open linked data related to omics experiments;

d) support the development of complementary data analyses;

e) develop a set of plugins (one per repository) to transform repository data into contextualised and linked information (knowledge) on demand. From the data collected in raw format, a transformation is performed to include contextual information (semantic metadata) based on the concepts and terms defined in the extended ontology.

The ontology shall:

a) use standard definitions and adopt terms from representative organisations;

b) be implemented using a mark-up language (see Annex B).

EXAMPLE Representative semantic models are: SAREF, Units of Measure, DCAT, and Schema.org.

Given the need to represent multi-omics data from different repositories in the more standardized way as possible, semantic models shall be constructed reusing, as much as possible existing (standardized and widely adopted) semantic models. The semantic models shall include terms and relations according to the data to be represented, providing a coherent and established structure for describing multi-omics data. This semantic representation shall facilitate the integration, interpretation, and automated analysis of experimental data sets from multiple sources.

In this way, the ISA model helps to:

a) resolve the complexity and diversity of multi-omics metadata;

b) provide a common standard for understanding and effective use of these data.

NOTE In the GLOMICAVE project we have used the ISA data model (see Annex A).

4.3.3 Validation of the semantic model

Semantic model validation is the process of verifying that the generated semantic model is suitable for the final use by the users and, therefore, can generate the relevant implicit knowledge for the users.

For the validation of the semantic model, it shall:

a) select a subset of data to be integrated and sufficiently relevant for the validation of the semantic model;

b) elaboration of the necessary tests to validate the model. To perform these tests, the tests shall be constructed using a knowledge graph query language;

c) construction of an intermediate knowledge graph to validate the contained knowledge;

d) documentation of the tests and the percentage of tests that fulfil the defined conditions (percentage of tests fulfilled);

e) continuous improvement and updating of the semantic model considering the results of the executed tests.

4.3.4 Documentation and publication of the ontology

The documentation and publication of the ontology is the process by which the definition and uses of the semantic model are exposed to the users to ensure its use and re-use. The ontology documentation process is an essential step for the continuity and continuous improvement of the ontology model; therefore, its publication is recommended so that the semantic model can be reused.

For ontology documentation, a tool should be used that:

a) automatically generates documentation based on the tags defined in the semantic model;

b) allows the integration of technologies that facilitate ontology processing, such as reasoning engines or semantic web;

c) allows accessibility to be guaranteed.

The ontology provides a digital structure that should allow information to be shared and explored using a common data exchange format based on standards.

EXAMPLE Standardized data formats such as: JSON-LD, RDF/XML, TTL, etc.

NOTE In the GLOMICAVE project, WIDOCO (https://github.com/dgarijo/Widoco) has been used for the generation of documentation, Ontology (https://ontoology.linkeddata.es/) for the integration and publication of ontologies and following the W3C criteria (https://www.w3.org/) to ensure accessibility (https://w3id.org/def/glomicave).

4.4 Knowledge graph construction and maintenance

4.4.1 Knowledge graph construction

The methodological procedure to build the knowledge graph shall have the following phases:

a) data acquisition from repositories (plugins);

b) ontology extension and construction; and

c) mappings and construction of the knowledge graph based on the information stored in a relational database.

NOTE 1 These steps cover the representation of data acquired from repositories and the subsequent transformation of this data to the semantic model. This transformation and final construction of the knowledge graph serve to improve the understanding of the data and its provenance. Subsequently, the construction of the knowledge graph facilitates data harmonization thanks to the adoption of a common terminology and agreed contextual information. Thus, the ontology used catalogues the entities involved to define the metadata and their relationships.

Once the necessary data are obtained from all repositories and stored in a relational database, the knowledge graph is built. To do this, the relevant data shall be extracted from the relational database by means of SQL queries, combining the data from different tables of the relational database according to the relationships previously defined in the ontology.

Once the relevant data has been obtained, the data shall be transformed by mapping it to the ontology. This process consists of relating the extracted data to the entities and properties defined in the ontology, ensuring that the data conforms to the structure of the ontology. To carry out this mapping process, graph modelling tools such as RDF shall be used to represent/serialize the data in the form of triples (subject, predicate, object), where each entity and semantic relationship is represented as a node in the graph. Once the data are structured according to the ontology structure, it shall be stored in a knowledge graph database.

NOTE 2 In the GLOMICAVE project, once the semantic data represented in triples were generated and stored in files with NT extension, it was decided to load and store these files in *TriplyDB*, which is a semantic database that allows to integrate these files in a knowledge graph compatible with the standards, and that allowed to publish this graph for its consultation and visualization.

To perform queries on the stored data, a semantic database query language such as SPARQL shall be used, which allows complex queries on knowledge graphs.

4.4.2 Knowledge graph maintenance

Knowledge graph maintenance is focused on ensuring that the Knowledge graph maintenance focuses on ensuring that the information contained in the knowledge graph is kept up to date and with the necessary quality to allow its study.

NOTE 1 The subject of knowledge graph maintenance, at present, is a branch that offers many challenges and opportunities because, the larger and more complex the graph is, the more difficult it is to maintain the interconnections between the knowledge and information contained.

To carry out the maintenance of the knowledge graph, it should:

a) maintain the data to knowledge transformation plugins;

b) maintain semantic models to ensure information quality and knowledge graph integrity; and

c) ensure the provenance and versioning of the information to track the information contained in the knowledge graph.

NOTE 2 In the GLOMICAVE project, the maintenance of the knowledge graph has been done through an iterative process to ensure the correct adequacy of the RML and SHACL models to ensure the quality of the knowledge graph.

Annex A
(informative)

ISA Data Model

The main entities in the ISA model are, as expected, the “research”, the “study” and the “assay”. The top-level “research entity” serves primarily as a container for one or more “studies”. Both “research” and “individual studies” may be linked to individuals, organizations, and publications. “Studies” represent a coherent unit of experimental work with a shared study design and set of experimental factors and describe the biological subjects of this work. “Studies” also contain one or more “assays,” which focus on the measurement process, related to the biological subjects in the study. “Assays” are limited to describing a particular type of measurement using a specific measurement technology, so multi-omics studies will always include multiple assays.

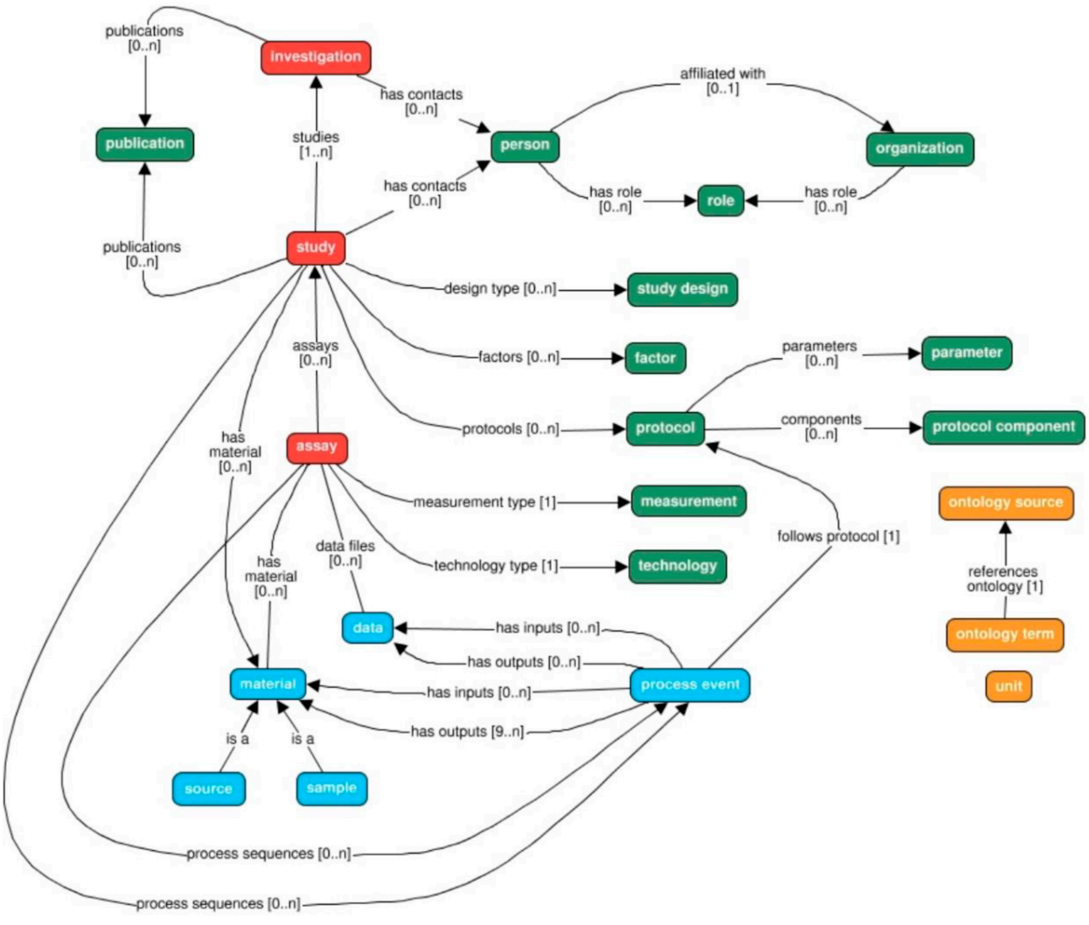


Figure A.1 — ISA data model

This schema, besides providing a representation of the entities to describe the repository data, also shows the relationships between them and how they are connected. The basis for the construction of the knowledge graph is to map the retrieved data with its representation within the schema.

Annex B
(informative)

Semantic model used in the GLOMICAVE project

The construction of the knowledge graph is presented by adapting and extending a pre-existing semantic data model called ISA (Investigation, Study, Essay). The elaborated approach and the semantic model (ontology) are publicly available at the following URL: https://colodidac.github.io/final-glomicave/OnToology/isa-glomicave.ttl/documentation/doc/index-en.html

Ontology implementation

The implementation language selected for the ontology is OWL (Ontology Web Language). The serialization format has been realized using Turtle language (TTL) and JSON-LD serialization to expose the information and corresponding variables acquired and stored in the different data sources such as semantic repository and unstructured databases. In addition, SPARQL has been selected as the semantic query language to retrieve and manipulate the data stored in the serialization formats (aforementioned) for the representation of the knowledge graph.

Ontology

The ontology is made up of a series of concepts defined in a specific context and how they relate to each other. For this purpose, a key step is the description of the entities represented in the previous scheme and their subsequent implementation in OWL format.

It is built around a set of key classes and their relationships with each other. It is structured around the concepts of research, study and test as main classes, which are the core of the whole ontology, and are derived in the definition of subclasses related to them (contacts, publications, materials, sources, samples, etc.).

The implementation in OWL is defined by the following namespaces:

Table B.1

Prefixes	Namespaces
:	https://w3id.org/def/isa-glomicave/
dcterms	https://purl.org/dc/terms/
owl	https://www.w3.org/2002/07/owl#
rdf	https://www.w3.org/1999/02/22-rdf-syntax-ns#
xsd	https://www.w3.org/2001/XMLSchema#
rdfs	https://www.w3.org/2000/01/rdf-schema#
vann	https://purl.org/vocab/vann/
time	https://www.w3.org/2006/time#
sf	http://www.opengis.net/ont/sf#
om	http://www.ontology-of-units-of-measure.org/resource/om-2/

Prefixes	Namespaces
schema	https://schema.org/
saref	https://saref.etsi.org/core/
isaterms	https://purl.org/isaterms
glomicave	https://w3id.org/def/isa-glomicave/

Bibliography

[1] Peroni S. «A Simplified Agile Methodology for Ontology Development. In Proceedings of the 13th OWL: Experiences and Directions Workshop and 5th OWL reasoner evaluation workshop (OWLED-ORE 2016),» 2016. [En línea]. Available: https://w3id.org/people/essepuntato/papers/samod-owled2016.html