Grant Agreement No.: 101080718
Call: HORIZON-HLTH-2022-STAYHLTH-01-two-stage
Topic: HORIZON-HLTH-2022-STAYHLTH-01-05-two-stage
Type of action: HORIZON-RIA

BIO-STREAMS

D3.3 Knowledge Graph Configuration Manual

Revision: v.1.0

Work package	WP 3
Task	Task 3.6
Due date	31/10/2024
Submission date	31/10/2024
Deliverable lead	NVCR
Version	1.0
Authors	Nikos Alimpertis, Aris Gioutlakis, George Fiotakis, George Domalis, Dimitris Tsakalidis, Ioannis Livieris (NVCR); Eleni Georga (UOI); Marianna Panagiotidou (AINIGMA); Spingos Ioannis, Roumelas George (UNI)
Reviewers	Eleftheria Tsourlidaki (UNI); Eleftheria Vellidou (ICCS)

Abstract	This deliverable serves as a comprehensive guide for the design, configuration, and deployment of a Knowledge Graph (KG) within the BIO-STREAMS project. By providing a structured framework and detailed procedures, this manual enables stakeholders to set up and customize their own KG instances according to their specific data requirements.

Keywords	knowledge graph, semantic annotation, configuration manual,

DOCUMENT REVISION HISTORY

Version	Date	Description of change	List of contributor(s)
V0.1	23/07/2024	1st version of the ToC	Nikolaos Alimpertis (NVCR)
V0.2	30/08/2024	Added deliverable scope and structure	Nikolaos Alimpertis (NVCR)
V0.3	13/09/2024	Added section 2 and 3 about knowledge graphs and semantic annotation	Nikolaos Alimpertis (NVCR), Spingos Ioannis, Roumelas George (UNI)
V0.4	26/09/2024	Added infrastructure requirements	Nikolaos Alimpertis, Aris Gioutlakis, Ioannis Livieris, Dimitris Tsakalidis (NVCR)
V0.5	08/10/2024	Added implementation steps	Nikolaos Alimpertis, Aris Gioutlakis, George Fiotakis, George Domalis (NVCR)
V0.6	15/10/2024	Added query requirements	Marianna Panagiotidou (AINIGMA)
V0.7	22/10/2024	Added the data model and the ontology	Eleni Georga (UOI)
V0.8	24/10/2024	Internal review	Eleftheria Vellidou (ICCS), Eleftheria Tsourlidaki (UNI)
V0.9	30/10/2024	Finalization for submission	Nikolaos Alimpertis, Aris Gioutlakis (NVCR)
V1.0	31/10/2024	Final version	Nikolaos Alimpertis, Aris Gioutlakis (NVCR)

Disclaimer

Copyright notice

Project co-funded by the European Commission in the Horizon Europe Programme		
Nature of the deliverable:	R	
Dissemination Level		
PU	*Public, fully open, e.g. web*	**X**
SEN	*Sensitive, limited under the conditions of the Grant Agreement*	

Classified R-UE/ EU-R	*EU RESTRICTED under the Commission Decision No2015/ 444*	
Classified C-UE/ EU-C	*EU CONFIDENTIAL under the Commission Decision No2015/ 444*	
Classified S-UE/ EU-S	*EU SECRET under the Commission Decision No2015/ 444*	

* *R: Document, report (excluding the periodic and final reports)*
DEM: Demonstrator, pilot, prototype, plan designs
DEC: Websites, patents filing, press & media actions, videos, etc.
DATA: Data sets, microdata, etc
DMP: Data management plan
ETHICS: Deliverables related to ethics issues.
SECURITY: Deliverables related to security issues
OTHER: Software, technical diagram, algorithms, models, etc.

Executive summary

This deliverable serves as a comprehensive guide for the configuration and setup of the Knowledge Graph (KG) for the BIO-STREAMS project. KGs play a crucial role in managing and querying complex datasets, providing a structured framework that enhances data interoperability and accessibility.

The document outlines the necessary steps for establishing an efficient KG environment that supports Resource Description Framework (RDF)-based data representation and facilitates semantic queries. It details the design principles and deployment strategies necessary for implementing KGs that effectively meet the requirements of semantic annotation.

For those interested in building their own KG, this configuration manual provides clear instructions for setting up a KG instance tailored to specific data subsets. By following these guidelines, users can customize their KG to fit their unique data management needs, ensuring they can effectively harness the power of semantic technologies.

Overall, this report is a valuable resource for users looking to enhance their data management capabilities through the effective use of KGs.

Table of contents

List of tables

Abbreviations

AE	Adverse Events
APDM	Associated Persons Demographics
APMH	Associated Persons Medical History
APPR	Associated Persons Procedures
APRP	Associated Persons Reproductive System Findings
APSC	Associated Persons Characteristics
APVS	Associated Persons Vital Signs
CM	Concomitant Medications (a domain in SDTM)
CSV	Comma-Separated Values
CDISC	Clinical Data Interchange Standards Consortium
DM	Demographics
EHRs	Electronic Health Records
EMA	European Medicines Agency
EX	Exposure (a domain in SDTM)
FDA	U.S. Food and Drug Administration
GOC	General Observation Classes
GDPR	General Data Protection Regulation
HTML	HyperText Markup Language
IS	Immunogenicity Specimen Assessments
JRE	Java Runtime Environment
JSON	JavaScript Object Notation
KG	Knowledge Graph
LB	Lab Results (a domain in SDTM)
ML	Machine Learning
MedDRA	Medical Dictionary for Regulatory Activities
MH	Medical History
PIP	Preferred Installer Program
py	Python (file extension for Python scripts)
QS	Questionnaires
RDF	Resource Description Framework
RP	Reproductive System Findings
RS	Disease Response and Clinical Classification
SC	Subject Characteristics
SDTM	Study Data Tabulation Model
SPARQL	SPARQL Protocol and RDF Query Language
TDB	Triple Database
ttl	Turtle (a serialization format for RDF data)
VS	Vital Signs
XML	eXtensible Markup Language

Co-funded by the European Union

1 Introduction

This deliverable is part of the BIO-STREAMS project and focuses on the design, configuration, and deployment of a Knowledge Graph (KG) aimed at supporting semantic annotation and facilitating data mapping. The KG plays a central role in improving the semantic integration, organization and interoperability of complex health data related to childhood and adolescent obesity. It enables efficient and accurate data mapping across multiple datasets, helping stakeholders like researchers, policymakers, and healthcare professionals gain meaningful insights.

Through the detailed steps provided in this manual, stakeholders can set up and configure their own KG instance for semantic annotation, adapting it to their specific data needs. By making health data discoverable, interconnected, and standardized, the KG allows users to perform advanced analyses and generate actionable insights. These insights will help in identifying patterns related to obesity prevention, intervention strategies, and healthcare decisions.

This document offers comprehensive technical guidance on configuring the KG, from the initial setup to its deployment, ensuring that all necessary steps for semantic enrichment and data interoperability are met. Designed with flexibility, the manual allows users to customize the KG with the data subsets that are most pertinent to their specific applications within the BIO-STREAMS framework.

1.1 Deliverable Scope

This deliverable (D3.3), titled "Knowledge Graph Configuration Manual," corresponds to Task 3.6, "Knowledge Graph and Semantic Annotation for Data Mapping," within WP3. It aims to provide a comprehensive guide for setting up and configuring a KG to enable semantic annotation and support data mapping within the BIO-STREAMS platform.

It builds upon the foundation set in D3.2, which introduced the common data model and ontology essential for data quality checks and harmonization. By leveraging these elements, D3.3 enhances the effectiveness of the KG in promoting accurate data integration and interoperability across various sources.

The manual is designed to assist developers and system architects in setting up the KG to enable efficient data mapping, analysis, and interoperability. It also provides stakeholders with an understanding of how the KG facilitates semantic annotation, helping streamline data integration across multiple sources.

The document outlines the KG’s design framework, configuration steps, and technical guidance, showcasing how the KG can enhance data accuracy and efficiency within the BIO-STREAMS platform.

1.2 Deliverable Structure

The document is organized as follows:

Section 2 introduces KGs by providing a comprehensive overview of their core concepts, including their relationship with ontologies. It highlights the benefits of KGs, particularly their role in enhancing data mapping and interoperability within the BIO-STREAMS platform. Additionally, relevant applications are presented to illustrate their practical importance, alongside key terminology and concepts referenced throughout the document.

Section 3 focuses on semantic annotation, offering an in-depth overview of this process and its critical role in ensuring accurate data mapping.

Section 4 outlines the design of KGs for semantic annotation, detailing key design principles, requirements, and objectives that shape the development of the KG in the context of BIO-STREAMS. This section covers the BIO-STREAMS common health data model and the corresponding ontology.

Section 5 provides guidance on the configuration and setup of the BIO-STREAMS KG, describing the steps required to configure and initialize a KG instance. It includes a detailed, step-by-step guide covering infrastructure requirements, tools, and configuration procedures, ensuring developers and system architects can successfully set up their own instance.

Section 6 presents the query requirements, focusing on methods for querying the KG to retrieve semantically annotated data. It covers the structure of SPARQL queries and includes examples of queries aligned with the platform's data mapping needs, as identified through stakeholder input. Additionally, it includes the best practices for optimizing queries and integrating results into the BIO-STREAMS platform.

Section 7 concludes the document by summarizing the key points discussed, emphasizing the role of KGs in supporting semantic annotation and data mapping within BIO-STREAMS. It reflects on the challenges encountered during the design and configuration process and explores potential areas for future development and refinement.

Co-funded by the European Union

2 Knowledge Graphs

2.1 Overview of Knowledge Graphs

KGs serve as vital tools for organizing and representing complex health information in various applications, for example in contexts such as childhood obesity. They employ a graph-based structure, where nodes represent entities (e.g., patients, medical records, interventions) and edges illustrate the relationships (of various types) among these entities (e.g., associations between symptoms, treatments, and outcomes).

In the BIO-STREAMS project, KGs facilitate the integration of diverse health data sources, including clinical records, genetic data, epidemiological studies, lifestyle factors, and social determinants of health. This structured representation is crucial for understanding the multifaceted nature of childhood obesity and related health challenges. By leveraging linked data principles, KGs enable meaningful connections between various data sources, supporting advanced analytics and predictive modeling across health domains.

A significant advantage of KGs is their ability to define and navigate relationships among medical, behavioral, and social factors through ontologies. This capability enhances the accurate representation of health statuses and increases the potential for predictive analysis regarding health outcomes and intervention effectiveness.

In summary, KGs, within the BIO-STREAMS project, represent a dynamic and interconnected resource that supports ongoing research and public health initiatives. They help uncover hidden patterns and inform the development of personalized interventions aimed at preventing and treating childhood obesity.

2.2 Ontologies and Knowledge Graphs

An ontology forms the foundational framework for any KG, defining the vocabulary that describes various entities (e.g., patients, treatments, clinical indicators) and the relationships that can exist between them. In the context of the BIO-STREAMS project, ontologies are essential for structuring and interpreting data related to childhood obesity. For example, an ontology may define key entities such as "patient", "vital signs" and "medical history" while elucidating the interrelations among these elements within the broader health context.

The ontology-driven approach ensures consistency and fosters a shared understanding of data throughout the BIO-STREAMS ecosystem. By establishing a common semantic framework, ontologies facilitate the meaningful integration of data from diverse sources, including healthcare providers, research institutions, and public health organizations. This integration is vital for generating comprehensive insights into the multifaceted nature of childhood obesity. Moreover, utilizing standardized vocabularies alongside project-specific ontologies enhances the KG's ability to support advanced queries and facilitate reasoning.

Additionally, ontologies empower reasoning capabilities within the KG, enabling the automatic inference of new knowledge from existing data. For instance, by analyzing relationships among genetic factors, family history, and obesity risks, the KG can identify potential high-risk groups. This capability significantly enhances the analytical power of the project, revealing insights that may be difficult to discern through traditional data analysis methods.

In conclusion, ontologies are fundamental to the functionality and effectiveness of KGs. They provide a structured approach to data representation that promotes interoperability, consistency, and advanced analytical capabilities.

Co-funded by the European Union

2.3 Benefits and Limitations of Knowledge Graphs

KGs can offer significant advantages in addressing the complex, multi-dimensional nature of childhood obesity and health data integration within the BIO-STREAMS project. One of the primary benefits of KGs is their ability to integrate structured, semi-structured, and unstructured data into a unified framework [1]. This integration enables the discovery of previously unknown relationships among medical, behavioral, and social factors affecting childhood obesity. By synthesizing data from clinical studies, electronic health records (EHRs), lifestyle factors, and social determinants of health, KGs provide a holistic view of individual and population-level health trends.

Additionally, the implementation of standardized ontologies and semantic web technologies enhances data interoperability [2, 3], simplifying the sharing and integration of information across diverse healthcare systems, research initiatives, and public health databases. This harmonization fosters collaboration among healthcare providers, researchers, and policymakers, enabling them to derive meaningful insights from a broader data spectrum [4]. For instance, linking clinical data on childhood obesity with socioeconomic factors or dietary patterns can unveil key drivers of health outcomes, supporting the design of more effective community-based interventions.

KG can contextualize health data by representing the sequence of events leading to specific health outcomes and mapping relationships among various risk factors. They can identify clusters of patients with similar health profiles, which is essential for understanding the multifactorial causes of obesity and evaluating the impacts of different interventions across diverse population groups. Moreover, the visualization capabilities of KGs aid policymakers and health professionals in monitoring intervention performance and effectively allocating resources.

However, despite these powerful capabilities, KGs face several challenges and limitations. Building and maintaining a KG may necessitate the integration of data from various and often disparate sources. In the context of BIO-STREAMS, this involves gathering information from different healthcare systems, behavioral studies, and demographic datasets, each with unique formats, granularity, and quality issues. Significant efforts are required for data cleaning, ontology alignment, and semantic consistency, which can complicate and prolong the construction of a functional KG.

As the KG expands in scope and scale, scalability and query performance become critical concerns. Health data, especially regarding childhood obesity, is dynamic, with continuous generation and updates of new datasets. Consequently, the KG must adapt and expand while maintaining efficient query performance. Querying large-scale health KGs with complex questions, such as assessing the long-term impacts of specific interventions on different population segments, requires robust infrastructure and indexing strategies to ensure timely responses, therefore improving clarity and fostering greater trust in the system.

Moreover, even the most well-structured KG cannot capture every detail or nuance of health domains, which can lead to potential gaps in knowledge. Incomplete or inconsistent data may result in misleading insights or ineffective interventions, particularly when addressing sensitive health issues like childhood obesity. Additionally, the inherent uncertainty in health data poses significant challenges for reasoning and decision-making. KGs shouldn't be static and must undergo continuous updates to remain relevant.

In summary, despite some limitations, KGs are invaluable tools within the BIO-STREAMS project for organizing and extracting insights from health data. Ongoing research and development efforts aim to tackle challenges related to scalability, data integration, and uncertainty, thereby enhancing the effectiveness of KGs in supporting health decision-making and intervention strategies.

2.4 Importance of Knowledge Graphs in Data Mapping

KGs play a pivotal role in data mapping by providing a structured framework to integrate and represent complex health data. Within the BIO-STREAMS project, KGs facilitate the linking of diverse datasets, as discussed in Section 2.1, enabling the identification of relationships among key health entities, including patients, treatments, and risk factors. By leveraging their graph-based structure, KGs streamline data mapping processes, which is essential for uncovering insights and understanding the multifaceted nature of childhood obesity.

A significant advantage of using KGs for data mapping is their ability to provide enriched contextual information about the data through the integration of different data sources. This contextualization is particularly important in health data, where the implications of relationships can substantially influence understanding and decision-making. For example, by mapping clinical data against behavioral and social factors, KGs can unveil hidden patterns that inform public health strategies and interventions.

Furthermore, KGs enhance interoperability among various health data sources. By utilizing standardized ontologies and vocabularies, KGs ensure that data from different origins can be integrated in a meaningful manner, facilitating comprehensive analyses. This interoperability is essential for collaborative research efforts, where multiple stakeholders need to share and analyze data from disparate systems. The capability to perform advanced queries across integrated datasets enables researchers and policymakers to derive actionable insights that may remain obscured when analyzing isolated datasets.

Moreover, KGs enable stakeholders to better visualize data relationships, making complex data more digestible. By representing data as interconnected nodes and edges, KGs provide intuitive visualizations that assist researchers, healthcare professionals, and policymakers in grasping the intricacies of the data landscape. This enhanced understanding can lead to more informed strategies and policies aimed at tackling childhood obesity effectively.

In conclusion, KGs are indispensable in the data mapping process within the BIO-STREAMS project. They not only facilitate the integration and contextualization of complex health data but also enhance interoperability and adaptability. By supporting advanced queries and enabling the visualization of relationships, KGs empower stakeholders to make well-informed decisions, ultimately leading to more effective interventions in the fight against childhood obesity.

2.5 Applications of Knowledge Graphs in BIO-STREAMS

The BIO-STREAMS KG serves as a foundational tool in multiple use cases, enabling efficient data integration, semantic annotation, and advanced Machine Learning (ML) capabilities. The KG supports various applications within the BIO-STREAMS project. One significant module where the KG plays a pivotal role is ML for Risk Assessment.

In the ML for Risk Assessment module, the KG is employed to develop an ML-based tool that predicts childhood obesity and associated health risks. The structured data within the KG enables the ML model to identify complex patterns and relationships, improving the accuracy of predictions. The KG integrates various data sources, such as clinical records, genetic data, behavioral information, and social determinants of health, allowing the ML model to draw more comprehensive and informed conclusions. By providing context and structure to the data, the KG enriches the ML process, leading to more accurate risk predictions and better decision-making in public health interventions.

Beyond this ML application, the KG supports other critical use cases within the BIO-STREAMS project. It enables seamless data sharing and collaboration among healthcare providers, researchers, and public health organizations by providing a unified framework for data interoperability. This ensures that data can be easily integrated, shared, and analyzed across

diverse platforms and institutions. Additionally, KGs support longitudinal data analysis, allowing researchers to track health outcomes over time and evaluate the long-term effects of various interventions. Visualization capabilities further enhance the utility of KGs by allowing stakeholders to explore data relationships in an intuitive and comprehensible manner.

Through these applications, the BIO-STREAMS KG is an essential tool that facilitates advanced data analysis, ML, and evidence-based decision-making in the fight against childhood obesity.

2.6 Terminology and Concepts

The foundation of KGs is built upon several key terms and concepts that are essential for understanding, constructing, and utilizing KGs to represent and leverage complex information in a structured and interconnected manner. Below, we outline some of the fundamental terms commonly associated with KGs, as referenced in this deliverable.

Table 1 – Terminology and Concepts in Knowledge Graphs

Term	Description
Entity	An entity represents a specific object, concept, or instance within the KG. Entities can be tangible objects (e.g., a person, a city) or abstract concepts (e.g., a domain concept, a class). In this project, each dataset is treated as a distinct entity.
Attribute	Attributes are properties or characteristics associated with entities in the KG, providing additional information. For instance, a person entity may have attributes like name, age, and occupation. Metadata describing each dataset is treated as attributes.
Relationship	Relationships define the connections or associations between entities in the KG, representing the semantic links and dependencies. Datasets can be associated based on various similarity metrics (e.g., “Associated by title” or “Associated by description”).
Classes and Types	Classes and types categorize entities into groups based on shared characteristics or attributes. In ontologies, classes represent concepts or categories within a specific domain.
Taxonomies	Taxonomies are hierarchical structures that organize entities into parent-child relationships based on similarities and differences, providing a way to classify entities into broader and more specific categories.
Triplet	A triplet is the fundamental building block of a KG, representing a single piece of information. It consists of three components: subject, predicate, and object. The subject and object are entities, while the predicate denotes the relationship between them. Note that the metrics used to establish triplets may not always be symmetrical; hence, the relationship is defined using Max(fun(A,B), fun(B,A)) to maintain the association's symmetry.
Graph	A graph is a collection of entities, attributes, relationships, and triplets organized in a network structure. It illustrates the overall structure and

	connections within the KG, with nodes representing entities and edges representing relationships.
Ontology	An ontology defines the schema or vocabulary for the KG, specifying the types of entities, relationships, and attributes that can exist. It provides a formal representation of domain knowledge, ensuring consistency and interoperability within the graph.
Ontology Hierarchies	Ontologies often include hierarchies that define relationships between classes and types, establishing a structured and logical organization of knowledge within the graph.
Domain-Specific Vocabulary	KGs frequently incorporate domain-specific vocabularies and terminologies relevant to the specific field of study. This ensures consistency and clarity in data representation.
Data Sources	KGs encompass information about the sources of data, such as satellite missions, data providers, and acquisition methods. This metadata helps users trace the origin of the information.
Semantic Annotations	Annotations within the ontology provide additional context or metadata about entities, relationships, or classes. Semantic annotations enhance the understanding and interoperability of data, facilitating better data integration and analysis.

3 Semantic Annotation

3.1 Overview of Semantic Annotation

Semantic annotation is the process of enriching data with metadata that provides contextual meaning to its content, allowing for more effective integration, retrieval, and analysis of information. In the context of the BIO-STREAMS project, semantic annotation plays a crucial role in ensuring that diverse health datasets can be understood in a unified manner by tagging the data with relevant concepts and relationships from domain-specific ontologies [5].

Through the use of semantic annotation, raw health data, including clinical records, genetic data, and lifestyle factors, can be associated with controlled vocabularies, making the data more interoperable across various platforms and systems. This structured approach not only facilitates data discovery but also enhances the ability to perform meaningful analysis across disparate datasets, helping to uncover new insights into childhood obesity and its underlying factors [6].

By assigning semantic labels to data, the annotation process enables KGs to map various concepts [7] and their interrelations, supporting sophisticated queries and advanced reasoning. This is particularly useful in multidisciplinary fields like healthcare, where data often originates from multiple sources and formats [5]. Semantic annotation acts as the bridge that connects these varied datasets, ensuring consistency and precision in how the information is interpreted and used [8].

3.2 Role of Semantic Annotation in Data Mapping

Semantic annotation is integral to the data mapping process within the BIO-STREAMS platform, providing a semantic layer that facilitates the connection between heterogeneous datasets. This process ensures that data from different sources, whether it's clinical, genetic or behavioral, can be aligned and integrated meaningfully. Through the use of standardized ontologies and vocabularies [9], semantic annotation helps to maintain consistency across datasets, allowing for more accurate mapping and analysis.

In data mapping, semantic annotations ensure that each piece of information is linked to a well-defined concept, making it easier to compare data points from different datasets. For instance, in the context of childhood obesity, differentially expressed concepts like "BMI," "physical activity," or "dietary intake" found in different datasets can be semantically associated despite differences in terminology or structure. By doing so, data mapping becomes more coherent, enabling advanced analytics that can better inform prevention and intervention strategies.

This consistency in data mapping also promotes data reusability. Once a dataset is semantically annotated, it can be more easily integrated with other datasets or reused for different research purposes, leading to more comprehensive studies. For policymakers and healthcare professionals, the ability to seamlessly map and integrate data across sources provides a powerful tool for evidence-based decision-making, enabling more effective targeting of public health interventions.

4 Design of Knowledge Graphs for Semantic Annotation

4.1 The BIO-STREAMS Common Health Data Model

The Clinical Data Interchange Standards Consortium (CDISC) [10] has established the Study Data Tabulation Model (SDTM) [11] as a comprehensive framework designed to standardize the organization and submission of clinical trial data to regulatory authorities, such as the U.S. Food and Drug Administration (FDA) [12] and the European Medicines Agency (EMA). The primary objective of SDTM is to facilitate efficient data exchange, promote data consistency, and ensure transparency in the analysis and reporting of clinical trial results.

SDTM provides a structured format for presenting clinical trial data in a uniform manner, organized into standardized domains or datasets. Each domain represents a specific category of clinical trial data, such as demographics, adverse events, laboratory results, or treatment interventions. By categorizing data into these pre-defined domains, SDTM ensures consistency and clarity across studies, regardless of therapeutic area, study design, or data origin. SDTM datasets are divided into three main types:

1. *Trial Design Datasets* – These capture essential study design elements, including planned treatments, visits, and randomization.
2. *Subject-level Data* – This includes information about individual study participants, such as baseline characteristics, adverse events, and medical history.
3. *Associated Reference Data* – Contains controlled terminology and metadata required for interpreting and linking subject-level data.

The SDTM structure relies on a consistent, modular approach. Each dataset consists of a series of variables with pre-specified names and formats, following a tabular structure. The General Observation Classes (GOC) define the core structure for most of the SDTM domains. The three main GOC classes include:

- *Interventions:* Record treatments, procedures, or actions that are part of the study, such as medication administration (tracked in domains like CM for Concomitant Medications and EX for Exposure).
- *Events:* Capture significant occurrences during the study, such as Adverse Events (AE) or Medical History (MH).
- *Findings:* Represent data collected from tests and assessments, like Lab Results (LB) or Vital Signs (VS).

Each of these classes has a specific structure of variables that are adapted across different domains based on the type of data being collected.

CDISC SDTM relies heavily on Controlled Terminology, which includes a set of pre-defined, standard terms used to describe data. Controlled terminology ensures consistency in how clinical terms are reported, aiding in interpretation and cross-study comparisons. Examples include terms for adverse events (such as MedDRA for medical dictionary codes) or lab test names and units (as defined by the National Cancer Institute's Thesaurus).

An overview of the BIO-STREAMS retrospective metadata mapping to the CDISC SDTM is provided in the following Table.

Table 2 – CDISC SDTM Mapping to BIO-STREAMS Dataset

BIO-STREAMS Data Category	CDISC SDTM Domain(s)
Healthcare Indices	Visits – Subject Visits (SV) Subject Status (SS)
Demographics	Demographics (DM) Subject Characteristics (SC)
Medical Data	Vital Signs (VS) Disease Response and Clinical Classification (RS) Medical History (MH) Laboratory Test Results (LB) Subject Characteristics (SC) Reproductive System Findings (RP) Concomitant/Prior Medications (CM)
Family Data	Associated Persons Demographics (APDM) Associated Persons Procedures (APPR) Associated Persons Vital Signs (APVS) Associated Persons Medical History (APMH) Associated Persons Characteristics (APSC) Associated Persons Reproductive System Findings (APRP)
Blood Tests	Immunogenicity Specimen Assessments (IS) Laboratory Test Results (LB)
Behavioural data	Questionnaires (QS)
Other	Questionnaires (QS)

Adopting the SDTM standard offers several advantages. First, it promotes interoperability between various stakeholders, such as researchers, sponsors, and regulators, by ensuring that data are presented in a clear, consistent format. This standardization reduces ambiguity, improving data quality and interpretability. Second, SDTM facilitates the review process by regulators, allowing for more streamlined validation and analysis, which can potentially accelerate the approval process. Lastly, SDTM contributes to the reuse and integration of clinical trial data across studies and therapeutic areas, supporting larger meta-analyses and

Co-funded by the European Union

advancing public health research. Overall, the CDISC SDTM common data model represents a critical step towards harmonizing clinical trial data.

4.2 The BIO-STREAMS Health Data Ontology

To effectively transition into the development of a KG, it is essential to understand the structure and content of the underlying BIO-STREAMS ontology that serves as its foundation. The BIO-STREAMS ontology is aligned with the CDISC SDTM, ensuring that the KG is built upon a solid, standardized framework for representing biomedical concepts. The ontology goes beyond simply categorizing retrospective data; it establishes formal relationships between entities, thereby laying a robust groundwork for the creation of the KG. By doing so, it provides a well-defined semantic structure that facilitates the integration and analysis of complex biomedical data.

Within the KG context, the ontology acts as a blueprint, defining the semantic relationships that guide the connections between data entities. This guidance is crucial for enabling the graph to represent biomedical knowledge accurately and effectively. The relationships defined within the ontology ensure that the KG captures not only the entities themselves but also the intricate dependencies and interactions among them.

The ontology's formal encoding in RDF/XML [13] allows for seamless integration with graph-based databases and tools, ensuring consistency with the selected common data model (CDISC SDTM) while retaining the flexibility to adapt to evolving requirements.

The BIO-STREAMS ontology is constructed in Protégé [14], an open-source ontology editor that allows for comprehensive ontology creation and management. The ontology development process (as elaborated in D3.2), starting from constructing the ontology in Protégé to preparing it for practical use as a KG, emphasizes the continuous alignment with the CDISC SDTM throughout the process. The representation of classes and subclasses shows how different concepts are interconnected, forming a detailed map of biomedical knowledge. This visual verification aids in confirming that the relationships between entities are logically coherent and prepared for translation into a KG. The graphical tools in Protégé help enhance the clarity of these connections, making the BIO-STREAMS ontology accessible for further development into a KG framework. Thus, they provide a comprehensive overview of how the ontology forms the backbone of the KG.

5 Configuration and Setup of Knowledge Graphs

This chapter outlines the steps necessary to configure and set up the KG infrastructure for the BIO-STREAMS project. The process includes defining the required infrastructure, followed by a detailed description of the configuration steps to set up a local KG, ingest data, and query it using SPARQL.

5.1 Infrastructure Requirements

Before setting up the KG for BIO-STREAMS, certain infrastructure components must be in place to ensure efficient operation. These include both software dependencies and system configurations that facilitate the use of RDF-based graphs and semantic data queries.

5.1.1 Software Requirements

The following is required to configure the BIO-STREAMS KG environment:

- **Operating System**: The system should run on a Unix-based OS (Linux or macOS). Windows may also be used, but with specific adaptations.
- **Java 8 or Higher**: Java Runtime Environment (JRE) version 8 or higher is required for running Apache Jena Fuseki, which serves as the RDF store for the KG.
- **Python 3.7 or Higher**: Python is needed for scripting data ingestion and manipulation tasks. Libraries like rdflib are used for working with RDF data.
- **Apache Jena Fuseki**: This is the RDF server used to host the BIO-STREAMS KG. It is responsible for managing SPARQL queries and data storage.
- **Git (Optional)**: Although not mandatory, Git is recommended for version control to manage ontology updates, configuration changes, and collaboration with other team members.

5.1.2 RDF Store and Ontology

Once the infrastructure requirements are met, the following steps outline the process to configure and set up the KG for BIO-STREAMS:

- **Fuseki TDB2 Dataset**: Fuseki's native TDB2 engine will be used to persist RDF triples. This ensures efficient querying and data management.
- **BIO-STREAMS Ontology**: The ontology file (in OWL format) defines the structure, relationships, and semantics of the KG. It should be uploaded to the RDF store as part of the initial setup.

5.2 Configuration Steps

Once the infrastructure requirements are met, the following steps outline the process to configure and set up the KG for BIO-STREAMS.

5.2.1 Install Apache Jena Fuseki

1. Download Apache Jena Fuseki

a. Navigate to the Apache Jena Fuseki download page.
b. Download the latest stable version of Fuseki and extract the archive to a suitable location on your machine.

2. Set Up Environment Variables
   a. Add the Fuseki bin directory to your system's PATH environment variable for easy access to the command-line tools.
3. Verify Installation
   a. Run the following command to verify the installation:
      - *fuseki-server --version*

5.2.2 Create and Configure RDF Store

1. Create Project Directory:
   a. Set up a directory structure for your BIO-STREAMS project where configuration files and datasets will be stored:
      - *mkdir ~/biostreams_kg*
      - *cd ~/biostreams_kg*
2. Configure the RDF Store:
   a. Create a fuseki-config.ttl configuration file in your project directory:

Table 3 – Configuration File (tll)

```
@prefix :      <http://base/#> .
@prefix tdb2:  <http://jena.apache.org/2016/tdb#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ja:    <http://jena.hpl.hp.com/2005/11/Assembler#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix fuseki: <http://jena.apache.org/fuseki#> .

:service1 rdf:type fuseki:Service ;
    rdfs:label                      "TDB2 BioStreams Dataset" ;
    fuseki:name                     "biostreams" ;
    fuseki:serviceQuery             "query" , "sparql" ;
    fuseki:serviceUpdate            "update" ;
    fuseki:serviceUpload            "upload" ;
    fuseki:serviceReadGraphStore    "get" ;
    fuseki:serviceReadWriteGraphStore "data" ;
    fuseki:dataset                  :dataset .

:dataset rdf:type      tdb2:DatasetTDB2 ;
    tdb2:location "DB2" .
```

3. Start Fuseki with the Configuration:
    a. Use the following command to start the Fuseki server with the custom configuration:
        - *fuseki-server --config=fuseki-config.ttl*

5.2.3 Load the BIO-STREAMS Ontology

1. Download Ontology File:
    a. Obtain the BIO-STREAMS ontology file (RDF/OWL format) from the shared project repository or team members.
2. Upload Ontology to Fuseki:
    a. Use the Fuseki web interface (http://localhost:3030) to upload the ontology to the BIO-STREAMS dataset.
    b. Alternatively, load the ontology from the command line using:
        - *tdbloader --loc=DB2 /path/to/BioStreams_ontology.rdf*

5.2.4 Data Ingestion

1. Install Required Python Libraries:
    a. Install the rdflib library for RDF manipulation using the following command:
        - *pip install rdflib*
2. Prepare Data Ingestion Script:
    a. Create a Python script (ingest_data.py) to convert data into RDF triples based on the BIO-STREAMS ontology. Customize the script to match your specific dataset structure and attributes.
3. Ingest Data:
    a. Run the script to generate RDF triples and serialize them in a Turtle file:
        - *python ingest_data.py*

Table 4 – Ingestion Script (py)

```
from rdflib import Graph, Literal, Namespace, URIRef
from rdflib.namespace import RDF, XSD

# Define namespaces
BS =
Namespace("http://www.semanticweb.org/lampr/ontologies/2024/7/BioStrea
ms_ontology#")

def create_subject(graph, subject_id):
    subject_uri = URIRef(f"{BS}Subject_{subject_id}")
    graph.add((subject_uri, RDF.type, BS.Subject))
    return subject_uri

def add_demographic(graph, subject_uri, age, sex):
    dm_uri = URIRef(f"{BS}DM_{subject_uri.split('_')[-1]}")
    graph.add((dm_uri, RDF.type, BS.DM))
```

```
    graph.add((dm_uri, BS.AGE, Literal(age, datatype=XSD.string)))
    graph.add((dm_uri, BS.SEX, Literal(sex, datatype=XSD.string)))
    graph.add((subject_uri, BS.has_demographic, dm_uri))

# Main ingestion function
def ingest_data(data_file):
    g = Graph()
    # Add your data ingestion logic here
    # Example:
    # with open(data_file, 'r') as f:
    #     for line in f:
    #         subject_id, age, sex = line.strip().split(',')
    #         subject_uri = create_subject(g, subject_id)
    #         add_demographic(g, subject_uri, age, sex)
    return g

# Run ingestion
graph = ingest_data('your_data_file.csv')
graph.serialize(destination='output.ttl', format='turtle')
```

b. Upload the resulting Turtle file to Fuseki:
   - *fuseki-http --update --file=output.ttl --base=http://example.org/ localhost:3030/biostreams*

5.2.5 Querying the Knowledge Graph

1. SPARQL Query Execution:
   b. You can now query your KG using SPARQL. The following is an example to retrieve subject demographic data:

Table 5 – Query File (sparql)

```
PREFIX bs:
<http://www.semanticweb.org/lampr/ontologies/2024/7/BioStreams_ontolog
y#>

SELECT ?subject ?age ?sex
WHERE {
  ?subject a bs:Subject ;
           bs:has_demographic ?dm .
  ?dm bs:AGE ?age ;
      bs:SEX ?sex .
}
```

```
LIMIT 10
```

2. Running Queries:
    a. Run this query either through Fuseki's web interface or by using command-line tools, ensuring that the query results align with the data mapping objectives of BIO-STREAMS.

5.2.6 Visualizing the Knowledge Graph

Visualizing the BIO-STREAMS KG helps in understanding the relationships and patterns within the data. This section describes methods for visualizing the KG, including both built-in tools and external libraries that offer more advanced capabilities.

1. Built-in Visualization Options
    a. Fuseki's Built-in Visualization

Apache Jena Fuseki provides basic visualization through its web interface, which is limited but sufficient for small queries. You can:

- View query results in tabular format.
- Export results in various formats such as JSON, CSV, or XML for external visualization.

However, for more interactive graph visualizations, it is recommended to integrate with external tools.

    b. GraphViz Integration

For small subsets of RDF data, you can use the GraphViz library in Python to generate simple graph visualizations. Here's a sample script that creates a directed graph from a Turtle file:

Table 6 – GraphViz Integration Script (py)

```
from graphviz import Digraph
from rdflib import Graph

def visualize_rdf_subset(rdf_file, output_file, limit=100):
    g = Graph()
    g.parse(rdf_file, format='turtle')

    dot = Digraph(comment='BioStreams Graph')
    dot.attr(rankdir='LR')

    # Add nodes and edges
    seen = set()
    count = 0

    for s, p, o in g:
```

```
        if count >= limit:
            break

        # Add subject node
        s_label = str(s).split('#')[-1]
        if s_label not in seen:
            dot.node(s_label, s_label)
            seen.add(s_label)

        # Add object node if it's not a literal
        if not isinstance(o, Literal):
            o_label = str(o).split('#')[-1]
            if o_label not in seen:
                dot.node(o_label, o_label)
                seen.add(o_label)

            # Add edge
            p_label = str(p).split('#')[-1]
            dot.edge(s_label, o_label, p_label)
            count += 1

    dot.render(output_file, view=True)
```

This script reads an RDF Turtle file, extracts subject-predicate-object triples, and generates a simple graph where each node represents an entity, and each edge represents a relationship.

2. Web-based Visualization Solutions
    a. Using D3.js

To enable a more interactive web-based visualization, you can use D3.js. First, you need to extract data from the KG using SPARQL queries and convert the results to a JSON format compatible with D3.js.

Table 7 – D3.js Integration Script (py)

```
from SPARQLWrapper import SPARQLWrapper, JSON
import json

def query_to_d3_format(endpoint_url, query):
    sparql = SPARQLWrapper(endpoint_url)
    sparql.setQuery(query)
    sparql.setReturnFormat(JSON)
    results = sparql.query().convert()
```

```
    # Convert to D3 format
    nodes = []
    links = []
    node_ids = {}

    for result in results["results"]["bindings"]:
        # Process nodes and links based on query results
        # Example for subject-medication visualization
        subject_id = result["subjectId"]["value"]
        medication = result["medication"]["value"]

        # Add nodes
        if subject_id not in node_ids:
            node_ids[subject_id] = len(nodes)
            nodes.append({"id": subject_id, "type": "subject"})
        if medication not in node_ids:
            node_ids[medication] = len(nodes)
            nodes.append({"id": medication, "type": "medication"})

        # Add link
        links.append({
            "source": node_ids[subject_id],
            "target": node_ids[medication],
            "type": "takes"
        })

    return {"nodes": nodes, "links": links}
```

Then you need to create an HTML file to visualize the data:

Table 8 – D3.js Graph Visualization (HTML)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>BioStreams Graph</title>
    <script src="https://d3js.org/d3.v7.min.js"></script>
    <style>
        .node { stroke: #fff; stroke-width: 1.5px; }
        .link { stroke: #999; stroke-opacity: 0.6; }
    </style>
</head>
<body>
```

```
    <svg width="960" height="600"></svg>
    <script>
        const svg = d3.select("svg");
        const width = +svg.attr("width");
        const height = +svg.attr("height");

        d3.json("graph_data.json").then(function(graph) {
            const simulation = d3.forceSimulation(graph.nodes)
                .force("link", d3.forceLink(graph.links).id(d =>
d.id))
                .force("charge", d3.forceManyBody())
                .force("center", d3.forceCenter(width / 2, height /
2));

            const link = svg.append("g")
                .selectAll("line")
                .data(graph.links)
                .join("line")
                .attr("class", "link");

            const node = svg.append("g")
                .selectAll("circle")
                .data(graph.nodes)
                .join("circle")
                .attr("class", "node")
                .attr("r", 5)
                .attr("fill", d => d.type === "subject" ? "#ff7f0e" :
"#1f77b4");

            node.append("title").text(d => d.id);

            simulation.on("tick", () => {
                link
                    .attr("x1", d => d.source.x)
                    .attr("y1", d => d.source.y)
                    .attr("x2", d => d.target.x)
                    .attr("y2", d => d.target.y);

                node
                    .attr("cx", d => d.x)
                    .attr("cy", d => d.y);
            });
        });
    </script>
</body>
</html>
```

b. Using Apache Graph Explorer

- Install Graph Explorer
- Configure it to connect to your Fuseki endpoint
- Use its built-in visualization features

3. Python-based Visualization

a. Using NetworkX

For RDF data analysis and visualization using Python, the following script utilizes the NetworkX library to create a visual representation of an RDF graph:

Table 9 – NetworkX RDF Graph Visualization Script (py)

```
import networkx as nx
import matplotlib.pyplot as plt
from rdflib import Graph as RDFGraph

def visualize_with_networkx(rdf_file):
    # Load RDF graph
    rdf_graph = RDFGraph()
    rdf_graph.parse(rdf_file, format='turtle')

    # Convert to NetworkX graph
    G = nx.Graph()

    # Add edges from RDF triples
    for s, p, o in rdf_graph:
        s_label = str(s).split('#')[-1]
        o_label = str(o).split('#')[-1] if not isinstance(o, Literal)
else str(o)
        p_label = str(p).split('#')[-1]

        G.add_edge(s_label, o_label, label=p_label)

    # Create visualization
    plt.figure(figsize=(12, 8))
    pos = nx.spring_layout(G)
    nx.draw(G, pos, with_labels=True, node_color='lightblue',
            node_size=1500, font_size=10)
    nx.draw_networkx_edge_labels(G, pos,
edge_labels=nx.get_edge_attributes(G, 'label'))
    plt.title('BioStreams Graph Visualization')
    plt.show()
```

6 Query Requirements

As described in Chapter 5, one of the applications of the BIO-STREAMS KG is the development of a Risk Assessment tool that is focused on childhood obesity and overweight.

Building a risk assessment tool using the KG offers several advantages over utilizing the raw data for the training of the AI model. By leveraging the inherent structure, flexibility, and enriched insights offered by a KG, the risk assessment tool can be more powerful, precise, and adaptive than one built solely on raw data. In the next two chapters we present the requirements in order to fully exploit the KG and deliver an accurate and effective risk assessment tool.

6.1 Query Identification

In order to train an ML-based obesity-focused risk assessment tool using a KG built on children's health data, it is necessary to identify the appropriate indicators and to define specific queries that can extract the relevant features. These queries must extract diverse and interconnected data points essential for successful ML processes. The basic dataset to be extracted from the KG includes medical variables such as BMI and puberty status, blood test results, e.g., cholesterol, hormone level, as well as demographic factors like age, gender, and ethnicity. Additionally, the queries must be designed to retrieve more complex data points, such as genetic predispositions (e.g., genetic markers linked to obesity), behavioral data related to physical activity, diet and sleep, and other factors related to family history and socioeconomic factors.

In addition, queries capable of capturing longitudinal or time-series data are critical for the development of an effective risk prediction model. By extracting health measurements, such as BMI and physical activity, across different time points, the model will be able to capture trends or changes in a child's health status over time that directly affect the risk of developing obesity or overweight in the future.

To fully exploit the advantages that KGs offer, queries must target the complex relationships within the graph, such as connections between genetic markers, health conditions, and environmental influences. To this end, ontological reasoning based on the BIO-STREAMS ontology may be required to retrieve any inferred relationships and provide the model with the ability to recognize conditions or risk factors that are related based on underlying semantic hierarchies. For example, this reasoning will unveil the link between diabetes and other metabolic disorders.

Finally, taking into consideration the diverse population whose data will contribute to the creation of the KG and the subsequent risk assessment tool, context-specific queries must be identified to target specific sub-populations. These subgroups might include children from different geographic regions, socioeconomic backgrounds, or age groups. This allows for the extraction of features relevant to these particular subgroups, such as identifying children in low-income neighborhoods or those with a family history of metabolic diseases, which improves the model’s capacity to assess risk in context-specific situations.

6.2 Query Implementation

Having identified the relevant queries, the next step is to implement these queries to efficiently extract the required data while adhering to performance, security, and scalability best practices. To access the data in the KG, SPARQL, a semantic query language able of retrieving and manipulating data stored in Resource Description Framework (RDF) format, will be utilized.

SPARQL allows users to extract specific subgraphs, find connections between entities, and perform reasoning over ontological structures. It also supports pattern-based searches across nodes and edges, enabling complex queries like traversing hierarchies or discovering hidden relationships, making it essential for fully exploiting the data in KGs. To make it less challenging for non-technical users, visual query builders are available online to simplify SPARQL queries generation(e.g., GraphDB Workbench). These queries can be particularly difficult to implement since they often require the use of optional filters and aggregation functions to account for incomplete or variable data across different records.

In the case of handling temporal data, specific queries will have to be implemented to retrieve longitudinal datasets. SPARQL supports the ordering of results by date, making it possible to extract time-series data for indicators such as BMI, which is critical for identifying trends over time and predicting future health risks.

In addition, in order to handle relational data, queries must be formulated in a way that allows the traversal of complex relationships within the KG. For example, genetic markers for obesity can be retrieved by querying nodes that represent children and their associated genetic markers, filtering for markers that are linked to obesity through related entities in the graph. Ontological reasoning is also important in this context, where inference rules can be applied to retrieve implicit relationships, improving the quality and depth of the data used for model training.

Context-specific queries are implemented by filtering data by geographic location, age group, or socioeconomic status. For example, in order to assess subgroups in terms of exposure to high pollution levels, the query must include additional filtering based on environmental conditions. These types of queries ensure that the model incorporates contextual features, improving its ability to make accurate predictions for specific subgroups.

Finally, scalability and performance must be considered when implementing queries. As the KG grows in size, efficient indexing of frequently queried attributes (such as BMI and age) becomes essential for maintaining fast query response times. Security and privacy also play a critical role during the implementation phase. Queries should ensure compliance with privacy regulations such as GDPR by de-identifying any personal information, such as names and addresses, while still providing access to the necessary health data for model training.

Once we identify and implement this functionality, the KG will become a valuable source of structured and interconnected data that enables the development of an accurate and robust risk assessment model for childhood obesity.

The specific SPARQL queries implemented based on the requirements outlined in this section can be found in Appendix A, which provides practical examples and further insights into query design.

7 Conclusion

This deliverable, "Knowledge Graph Configuration Manual," presents a detailed guide for the setup, configuration, and deployment of KGs within the BIO-STREAMS project. KGs are crucial tools for managing complex datasets and improving semantic integration, particularly for health data related to childhood and adolescent obesity. Despite the progress made, several challenges emerged during the development process, and this conclusion reflects on those, provides a recap of key techniques, and discusses future prospects for applying the KG in real-world contexts.

7.1 Challenges and Mitigation Strategies

One of the main challenges encountered during the project was the absence of retrospective data, which limited the ability to fully apply and validate the manual during the project's development phase. Furthermore, the finalization of the ontology and data model occurred in the last stages of the deliverable, restricting early-stage testing. This delay underscores the challenge of synchronizing data model development with real-world data availability.

To address this, the manual was designed with flexibility, ensuring that it can be applied to real datasets in the future. The use of standardized frameworks and automated data ingestion processes will help ensure smooth integration once the data becomes available. Additionally, privacy concerns and scalability were addressed through built-in best practices and optimization strategies for query performance. Collaborative efforts with the partners will be essential in creating standardized queries and analysis pipelines for shared use.

7.2 Recap of Key Concepts and Techniques

This deliverable provides a detailed overview of essential concepts and techniques for implementing KGs to support semantic annotation and data mapping. Key to this is the design of the BIO-STREAMS ontology, aligned with the CDISC SDTM standard, which offers a structured and consistent approach for integrating diverse health datasets.

The manual includes step-by-step instructions for setting up the infrastructure, configuring Apache Jena Fuseki, ingesting data, and executing SPARQL queries. Guidance is also provided for handling complex relationships necessary for advanced health analysis, such as risk assessment models.

In addition, visualization methods via tools, allowing users to gain insight into the relationships between various health entities. This functionality supports the broader goal of fostering a deeper understanding of the factors contributing to childhood obesity.

7.3 Future Prospects and Further Exploration

Looking ahead, the application of the KG to real retrospectively or prospectively collected BIO-STREAMS datasets is a critical next step. This will enable a full assessment of the KG's utility in semantic annotation and complex data analysis. Future efforts will explore the creation of more advanced SPARQL queries tailored to specific research questions, particularly in understanding childhood obesity trends and interventions.

Further development of automated data ingestion pipelines will be explored, allowing the KG to handle regularly updated datasets efficiently. Collaborations with the partners will also be crucial, particularly in establishing shared queries and standardized analysis pipelines that ensure data interoperability and collaborative research across institutions.

In conclusion, the KG configuration manual outlined in this deliverable provides a strong foundation for future developments within the BIO-STREAMS project. By continuing to refine the data model, ontology, and querying capabilities, the KG will become an invaluable tool in BIO-STREAMS project.

8 References

[1] Wang Z., et al. (2014). Knowledge graph and text jointly embedding. In Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1591-1601.

[2] Malinverni E.S., et al. (2022). A semantic graph database for the interoperability of 3D GIS data. Appl Geomat, 14(Suppl 1), 53–66.

[3] Collarana, D., et al. (2017). Semantic Data Integration for Knowledge Graph Construction at Query Time. In 2017 IEEE 11th International Conference on Semantic Computing (ICSC), San Diego, CA, USA, pp. 109-116.

[4] Dilmegani, C. (2024). In-depth Guide to Knowledge Graph: Use Cases.

[5] Palojoki, S., et al. (2024). Semantic Interoperability of Electronic Health Records: Systematic Review of Alternative Approaches for Enhancing Patient Information Availability.

[6] Bodenreider, O., & Stevens, R. (2006). Bio-Ontologies: Current Trends and Future Directions.

[7] Ayatollahi, H. (2022). Health Data Science and Semantic Technologies. In Data Science with Semantic Technologies. Wiley Online Library.

[8] Iroju, O., et al. (2012). Semantic interoperability in healthcare: Motivating the critical need for ontology matching in healthcare. International Journal of Computer.

[9] Hoehndorf, R., et al. (2015). The role of ontologies in biological and biomedical research: A functional perspective.

[10] CDISC. "About CDISC." Available at: https://www.cdisc.org/about

[11] CDISC Study Data Tabulation Model (SDTM) v1.7. Available at: https://www.cdisc.org/standards/foundational/sdtm

[12] Food and Drug Administration. "Study Data Standards Resources." Available at: https://www.fda.gov/industry/fda-data-standards-advisory-board/study-data-standards-resources

[13] Protégé, A free, open-source ontology editor and framework for building intelligent systems. Accessed: Oct. 30, 2024. [Online]. Available: https://protege.stanford.edu/

[14] W3C Resource Description Framework (RDF). Accessed: Oct. 30, 2024. [Online]. Available: https://www.w3.org/RDF/

Appendix A: Aligned SPARQL Queries for BIO-STREAMS Data Structure

This appendix presents a collection of aligned SPARQL queries designed for the BIO-STREAMS Knowledge Graph. Each query targets specific aspects of the dataset, including lab results, medical history, treatment responses, and patient journeys. The queries have been crafted to efficiently extract relevant information while adhering to the structural conventions of the underlying data.

- Query 1: Find Subjects with Specific Lab Results and Their Demographics

This query retrieves subjects with high glucose levels along with their demographic information.

```
PREFIX bs:
<http://www.semanticweb.org/lampr/ontologies/2024/7/BioStreams_ontolog
y#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?subjectId ?age ?sex ?testValue
WHERE {
  ?subject a bs:Subject ;
           bs:has_demographic ?dm ;
           bs:Related_to ?lb .

  # Extract subject ID from URI
  BIND(STRAFTER(STR(?subject), "Subject_") AS ?subjectId)

  # Get demographic information
  ?dm a bs:DM ;
      bs:AGE ?age ;
      bs:SEX ?sex .

  # Get lab results
  ?lb a bs:LB ;
      bs:LBTEST "Glucose" ;
      bs:LBTESTCD "GLUC" ;
      bs:LBSTRESN ?testValue .

  # Filter high glucose values
  FILTER(xsd:float(?testValue) > 100)
}
ORDER BY DESC(xsd:float(?testValue))
```

- Query 2: Medical History Analysis with Vital Signs

This query analyzes patients' medical history in conjunction with their vital signs, focusing on cardiovascular issues.

```
PREFIX bs:
<http://www.semanticweb.org/lampr/ontologies/2024/7/BioStreams_ontolog
y#>

SELECT ?mhterm (COUNT(DISTINCT ?subject) as ?patientCount)
       (AVG(xsd:float(?bpValue)) as ?avgBP)
WHERE {
  ?subject a bs:Subject ;
           bs:Related_to ?mh ;
           bs:Related_to ?vs .

  # Get medical history
  ?mh a bs:MH ;
      bs:MHTERM ?mhterm ;
      bs:MHBODSYS "Cardiovascular" .

  # Get vital signs
  ?vs a bs:VS ;
      bs:VSTEST "Blood Pressure" ;
      bs:VSSORRES ?bpValue .

  # Extract systolic BP from combined BP value
  BIND(xsd:float(STRBEFORE(?bpValue, "/")) AS ?systolicBP)
}
GROUP BY ?mhterm
HAVING (?patientCount > 1)
ORDER BY DESC(?avgBP)
```

- Query 3: Treatment Response Analysis

This query evaluates the effectiveness of treatments across different age groups and visit numbers.

```
PREFIX bs:
<http://www.semanticweb.org/lampr/ontologies/2024/7/BioStreams_ontolog
y#>

SELECT ?treatment ?ageGroup ?visitNum (COUNT(DISTINCT ?subject) as
?subjectCount)
WHERE {
  ?subject a bs:Subject ;
           bs:Related_to ?cm ;
           bs:has_demographic ?dm ;
           bs:Related_to ?sv .

  # Get medication information
  ?cm a bs:CM ;
      bs:CMTRT ?treatment .

  # Get demographic information
  ?dm bs:AGE ?age .

  # Get visit information
  ?sv a bs:SV ;
      bs:VISIT ?visit ;
      bs:VISITNUM ?visitNum .

  # Create age groups
  BIND(
    IF(xsd:integer(?age) < 30, "Under 30",
      IF(xsd:integer(?age) < 50, "30-50",
        IF(xsd:integer(?age) < 70, "50-70", "Over 70")
      )
    ) as ?ageGroup
  )
}
GROUP BY ?treatment ?ageGroup ?visitNum
HAVING (?subjectCount > 0)
ORDER BY ?treatment ?ageGroup ?visitNum
```

- Query 4: Complex Patient Journey Analysis

This query consolidates information on patient visits, medical history, treatments, and lab tests into a comprehensive overview.

```
PREFIX bs:
<http://www.semanticweb.org/lampr/ontologies/2024/7/BioStreams_ontology#>

SELECT ?subjectId ?visit
       (GROUP_CONCAT(DISTINCT ?mhterm; SEPARATOR=", ") as ?conditions)
       (GROUP_CONCAT(DISTINCT ?treatment; SEPARATOR=", ") as ?medications)
       (GROUP_CONCAT(DISTINCT ?labTest; SEPARATOR=", ") as ?labs)
WHERE {
  ?subject a bs:Subject .
  BIND(STRAFTER(STR(?subject), "Subject_") AS ?subjectId)

  # Get visit information
  ?subject bs:Related_to ?sv .
  ?sv a bs:SV ;
      bs:VISIT ?visit .

  # Get medical history
  OPTIONAL {
    ?subject bs:Related_to ?mh .
    ?mh a bs:MH ;
        bs:MHTERM ?mhterm .
  }

  # Get medications
  OPTIONAL {
    ?subject bs:Related_to ?cm .
    ?cm a bs:CM ;
        bs:CMTRT ?treatment .
  }

  # Get lab tests
  OPTIONAL {
    ?subject bs:Related_to ?lb .
    ?lb a bs:LB ;
        bs:LBTEST ?labTest .
  }
}
GROUP BY ?subjectId ?visit
ORDER BY ?subjectId ?visit
```

These queries are now fully aligned with:

1. The TSV file structure.
2. The data loader implementation.
3. The RDF graph structure created by our loader.
4. The property names and relationships in our data.