Data in Brief 63 (2025) 112137

Contents lists available at ScienceDirect

Data in Brief

journal homepage: www.elsevier.com/locate/dib

Data Article

Structured RDF dataset for genomic feature extraction and detection of biotechnological microorganisms of the *Burkholderia* genus

Reynold Osuna-González[a,*], Guillermo De Ita Luna[a], Rosa María Valdovinos-Rosas[b], Yagul Pedraza-Pérez[c]

[a] *Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, Avenida San Claudio s/n, Ciudad Universitaria, La Hacienda C.P. 72592, Puebla, Puebla, Mexico*
[b] *Facultad de Ingeniería, Universidad Autónoma del Estado de México, Cerro de Coatepec S/N Ciudad Universitaria C.P. 50100. Toluca, Estado de México, Mexico*
[c] *Puebla, México.*

ARTICLE INFO

Article history:
Received 19 May 2025
Revised 23 September 2025
Accepted 29 September 2025
Available online 9 October 2025

Dataset link: Burkholderia Genomic RDF Graph (Original data)

Keywords:
Resource description framework
Burkholderia
Genomics
BGFF
Bioinformatics
Data representation

ABSTRACT

The current dataset has been built from 200 genomes of *Burkholderia* genus microorganisms in GenBank Flat File (GBFF) format, with information obtained from the National Center for Biotechnology Information (NCBI). Python scripts were developed to automate the extraction of genomic features such as coding sequences (CDS) and different types of RNA, extracting the descriptive information of each one. The extracted features were transformed into RDF (Resource Description Framework) format as initial knowledge graphs with Turtle syntax, and later, they were merged into a unified knowledge graph (KG) to facilitate their access via queries written in the SPARQL system.

This dataset is publicly available to researchers through the Mendeley repository, allowing them to perform complex searches, looking for composite features coming from the dataset's source organisms. Choosing the RDF format for the knowledge graph also promotes interoperability with

* Corresponding author.
E-mail address: reynold.osunag@correo.buap.mx (R. Osuna-González).

https://doi.org/10.1016/j.dib.2025.112137

other biological datasets that use the same structure and are widely accessible through SPARQL Endpoints.

Specifications Table

Subject	*Biology*
Specific subject area	*Genomic Information*
Type of data	Turtle RDF Graph
Data collection	*The dataset consists of 200 genomes of microorganisms from Burkholderia genus in GenBank Flat File (GBFF) format. The data were processed through Python scripts to extract relevant genomic features such as genes, coding sequences (CDS), transfer ribonucleic acid (tRNA), ribosomal ribonucleic acid (rRNA) and non-coding ribonucleic acid (ncRNA).* *The extracted data are structured into CSV files which allow to create RDF graphs with Turtle syntax, corresponding to each input genome.* *All RDF graphs were unified into a unique graph, enabling queries through SPARQL, and facilitating the integration of the information into other biological datasets.*
Data source location	To build the RDF graph, files from the National Center for Biotechnology Information (NCBI) https://www.ncbi.nlm.nih.gov/ were used. 200 genomes of microorganisms from genus *Burkholderia* were processed to build the RDF graph.
Data accessibility	All source scripts, intermediate files, and the final RDF graph are available in the public Mendeley Data repository: https://doi.org/10.17632/pt6xn9mgdf.6 All scripts are distributed under the Creative Commons Attribution License (CC BY 4.0), which allows reuse with appropriate credit to the authors.
Related research article	*None*

1. Value of the Data

- The advancement in DNA sequencing techniques has led to the generation of big volumes of genomic data. By identifying the nucleotide sequence that constitutes the DNA and determining the number of nucleotides that form each gene, a genome is obtained (a map of the genes of an organism). When an organism is sequenced for the first time and a related study is published, it is customary to make the genomic data publicly available through specialized databases. These databases have grown so fast that studying and analyzing the available information requires significant computational resources and time, making it prohibitive to conduct such analyses through traditional techniques [1].
- One application of molecular biology is the identification of useful metabolites that may lead to discover new drugs, as well as their use as pest control agents. In this process, when a useful metabolite is found, the genes responsible for its metabolism are identified and a search for organisms with the same or similar genes is made, trying to identify which one is capable to produce it. To infer an organism's capacity to metabolize a target metabolite, direct comparisons of gene or protein sequences among organisms within the same genus are essential. While an exact sequence match strongly suggests metabolic capability, partial gene matches or genomic similarities may also indicate functional potential, as some organisms can produce the metabolite even with incomplete gene sets associated with its biosynthesis. Thus, analyses should prioritize both exact matches and sequence similarities to avoid false-negative conclusions [2].
- The *Basic Local Alignment Search Tool* (BLAST) [3] is one of the most widely used algorithms to identify functional similarities between nucleotide sequences, as well as to infer the function of unknown sequences [4]. Applying BLAST to a large set of genomes, using the sequences

corresponding to the genes of interest, can become a slow process due to the sheer number of comparisons required and the need for opening and closing multiple files. "It is important to present the report as quickly as possible, but in some cases, formatting the alignments can delay loading of the page and can consume substantial resources on the user's desktop" [5].

- Although no formal benchmarking against BLAST was conducted, SPARQL queries on the RDF graph significantly reduce file I/O operations and enable rapid multi-feature searches that are difficult to replicate with BLAST. Unlike BLAST, which is designed to evaluate sequence similarity, the proposed RDF graph does not aim to perform approximate or alignment-based comparisons. Instead, it facilitates exact-match searches on annotated gene and protein features, particularly when a gene or protein already has an assigned name or locus_tag. In such cases, researchers can retrieve all genomes containing specific features (e.g., ssrA, gyrB) efficiently through semantic queries, without requiring sequence alignment or local installations.
- The steps followed in this work can be applied to the elaboration of RDF (Resource Description Framework) graphs [6] from any genomic information contained in GBFF (GenBank Flat File) format files [7], regardless of the organism or taxonomic group involved. The dataset described in this paper was constructed from genomic information of 200 microorganisms belonging to the genus *Burkholderia*. This selection is due to the fact that two authors collaborated with a research group that uses this genus as a study model. From this graph, descriptive information regarding genomic features can be retrieved efficiently, such as the matching of a defined set of genes, contributing to the rapid identification of candidate organisms for laboratory experimentation.
- The genomic information encoded into this knowledge graph is made available to researchers working with the *Burkholderia* genus. Through SPARQL queries, they can quickly search for protein sequences, genes, tags, gene names or taxonomy, among other features, facilitating the identification of organisms meeting specific search requirements.

2. Background

A GenBank Flat File (GBFF) is a type of file used to describe genomes, allowing for the enrichment of DNA sequences or subsections of it, such as chromosomes, plasmids and genes information, with descriptions such as names, functions, identifiers, and the translation of DNA into proteins [7].

The GBFF format allows the addition of free-text information, resulting in significant variability in the content of different genomes and subsections. This variability introduces extra complexity when creating a structured relational representation for multiple GBFF files, thereby making the information retrieval process, such as sequence matching, laborious and time-consuming.

A Resource Description Framework (RDF) graph is a structured representation of data where information is organized into triples: (subject, predicate, object), enabling the modeling of relationships between different entities. Thus, an RDF graph representation can be considered a non-relational database that facilitates the representation of hierarchical, taxonomic and multiple connections between vertices (Fig. 1).

Building a genomic information knowledge graph (RDF) enables the storage of hundreds of organisms' genomes in a single file. This file can be queried efficiently using the SPARQL query language. Moreover, the knowledge graph can be easily extended to incorporate new genomes.

3. Data Description

The dataset (Table 1) consists of an RDF graph constructed from genomic information belonging to the *Burkholderia* genus, along with the original GBFF files and the intermediate files generated during the processing stages. All these resources are available at [8] and are:

- Raw Data Folder: Contains all the 200 original GBFF files downloaded from the NCBI database.

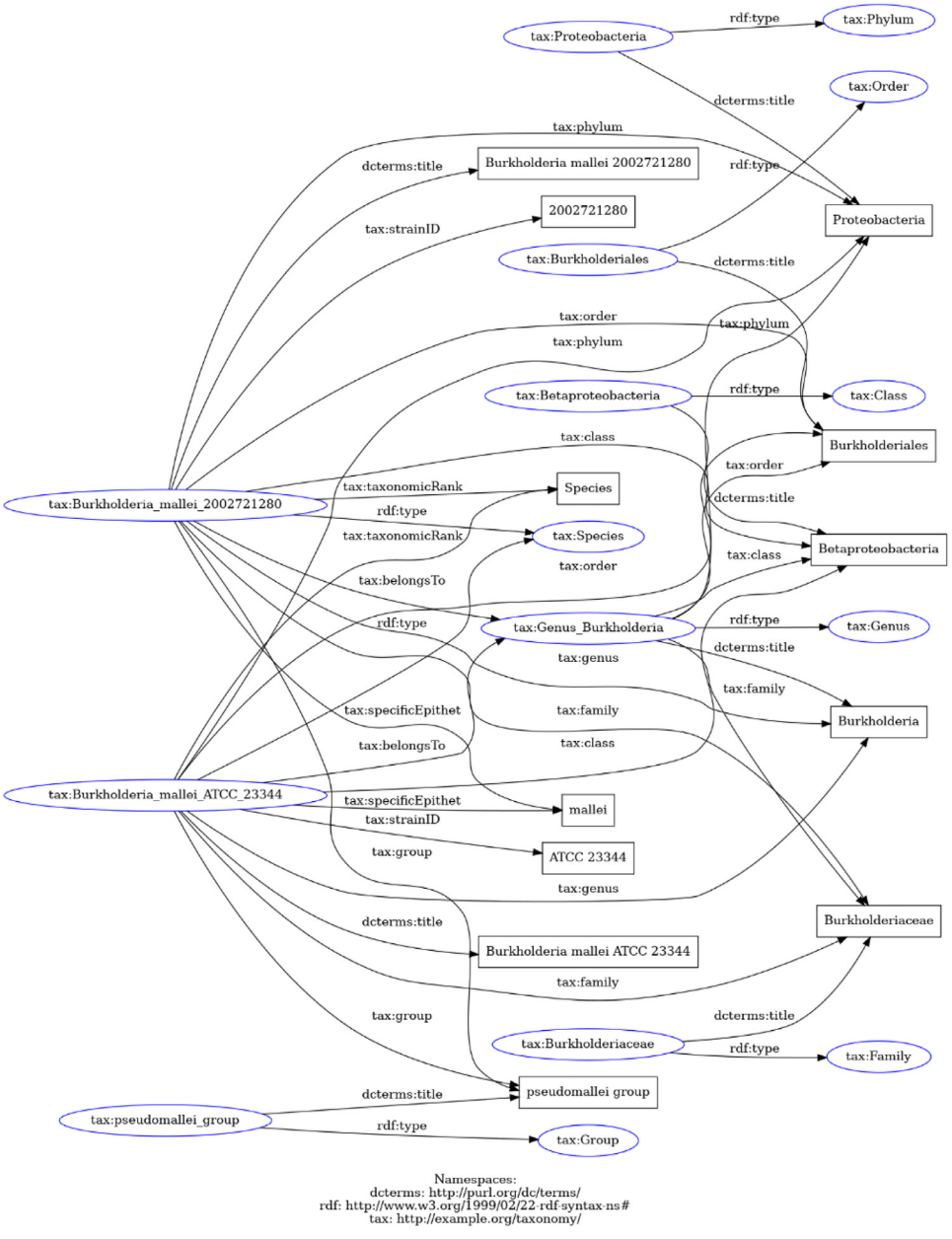


Fig. 1. Taxonomy of three distinct strains of Burkholderia mallei in RDF format.

- Preprocessed Data: ZIP file with 200 .txt files where unnecessary line breaks have been removed.
- Processed Data Folder: Contains 200 CSV files with the genomic features extracted from each genome.
- Individual RDF Graphs Folder: Contains the RDF graphs in Turtle (*.ttl) format, corresponding to the features extracted from each genome.

Table 1
Structure and content of the data in the repository.

File	Content	Purpose
Raw Data	200 original GBFF files	Original genomic data
Preprocessed Data	Compressed zip file with 200 TXT files without line breakers	Allows for the correct extraction of genomic features
Processed Data	200 CSV files with extracted genomic features	Structures genomic information into columns to support RDF graph construction
Individual RDF Graphs	200 TTL files, one per each original GBFF file	Genomic representation in RDF format for each organism
Unified RDF Graph File	A TTL file containing the merged information of all 200 organisms in RDF knowledge graph format	Enables SPARQL queries across all processed genomes

Table 2
Metrics of the resulting RDF Graph.

Metric	Value
Number of RDF triples	34,065,980
Total number of nodes	5521,453
Number of node types	2
– Locus_Segment nodes	3888
– Subsegment nodes	1287,225
Number of unique properties	392

- Unified RDF Graph File: A single Turtle (*.ttl) file containing the unified knowledge graph information from all 200 organisms.

The metrics of the unified RDF Graph resulting from the transformation of the 200 GBBF files can be found in table 2 below:

4. Experimental Design, Materials and Methods

4.1. Source data

To construct the knowledge graph (KG), files from the National Center for Biotechnology Information (NCBI) (https://www.ncbi.nlm.nih.gov/) [9] were used. This repository freely stores, analyzes, and distributes biological and biomedical information and currently contains over 2.8 million genomes from organisms belonging to the Eukaryota, Archaea and Bacteria kingdoms as well as viral genomes, provided in various genomic information file formats, including the GenBank Flat File (GBFF).

The GBFF files contain information about nucleotide sequences and metadata, following the Feature Table definition published by the International Nucleotide Sequence Database Collaboration (https://www.insdc.org/submitting-standards/feature-table/)[10]. It describes one or more nucleotide sequences, generically referred to as loci [11]: a chromosome contains multiple genes; however, both, the chromosome and a single gene are nucleotides identified as loci. Generally, each GBFF file contains the genome information of a microorganism, divided into large sequences such as chromosomes or plasmids and the smaller sequences they are composed of, such as genes. The description of each chromosome or plasmid is, in turn, divided into three sections:

- **LOCUS.** This section describes the chromosome or plasmid of the genome. It begins with the word “LOCUS” in uppercase letters, followed by its name and information about the type of molecule it describes, such as chromosomes, genes, coding sequences or plasmids (https:

//www.genome.gov/es/genetics-glossary/Locus)[10]. Optionally, it may include the organism's taxonomy and various metadata, as well as publication information regarding the original sequencing and sample origin.

- **Features.** Divided into two parts, this section contains detailed information about the LOCUS. The first part may describe the source material, the amino acid size of the LOCUS, the type of molecule, the microorganism strain, the isolation source, whether it describes a chromosome or a plasmid, and optionally, various metadata the file's author deemed relevant. The second part is divided into smaller loci descriptions (functional groups), each composed of three main fields:
  - *Feature key*: Functional group to which a subsequence may belong (e.g., source, gene, CDS, RNA, etc.).
  - *Location*: The base position of the locus within the sequence, allowing determination of nucleotide length.
  - *Qualifiers*: Descriptive information about the locus, which can vary greatly depending on the functional group and the author's discretion. However, for every CDS, the "translation" qualifier is always present, containing a character string representing the amino acids of the corresponding protein.
- **Origin.** This section includes the nucleotide sequence that composes the chromosome or plasmid, i.e., the DNA sequencing corresponding to the LOCUS.

4.2. Feature selection

Through a reverse-engineering approach, the expected types of SPARQL queries that researchers might use while exploring metabolic of taxonomic patterns were analyzed. This analysis led to the selection of features to be extracted, based on their frequency, biological significance and ability to be semantically represented in RDF format: Genes, CDS, rRNAs, tRNAs and ncRNAs. This approach contributes to the uniqueness of the dataset by aligning it with semantic interoperability goals and facilitating structured querying in genomic research. _

4.3. Knowledge graph creation

The steps for building the knowledge graph are divided into three stages, as shown in Fig. 2. ***Step 1. Data acquisition and Preprocessing.*** This stage comprises the following steps:

1. **Retrieve data:** The GBFF files providing source information for graph construction were obtained via direct download from the NCBI website (Search NCBI databases - NLM). In the Genomes section, the "Assembly" option was selected, and microorganisms belonging to the *Burkholderia* genus were retrieved.
   From the results, individual genomes could be selected for download. The "Download Package" option was chosen, and the file type "Sequence and annotation (GBFF)" was specified. This process resulted in a compressed (zip) file containing the folder ncbi_dataset/data, where the genomes were stored in individual folders named according to their GenBank identifier.
2. **Preprocessing:** The data folder was decompressed using the Files_renaming routine, which copied all files with the ".gbff" extension from the individual folders into a working directory, simultaneously renaming each file from "genomic" to the corresponding GenBank identifier (taken from its original folder name). Subsequently, the preprocessing routine was applied to clean the files, eliminating unnecessary line breaks to consolidate the data into a uniform format.
3. **Information Extraction:** In the LOCUS section, essential information was extracted to construct the knowledge graph, including the scientific name of the organism, its strain, and its

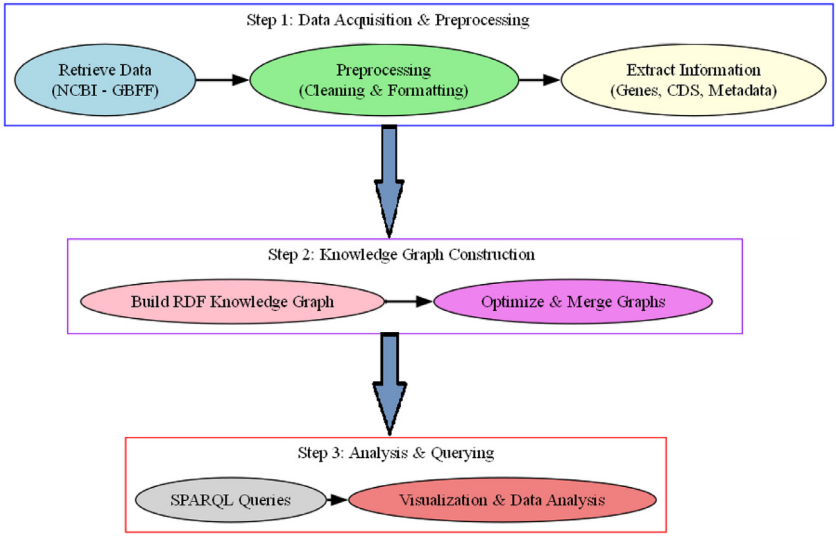


Fig. 2. Methodology for dataset creation and information extraction.

taxonomic identifier. As previously noted, the LOCUS section may or may not contain additional metadata.

The descriptions for each functional group may contain relevant data such as the locus_tag (unique identifier for the functional group), gene name (if available), or a functional description, along with any annotations that the sequencing author deemed important. The routines required for genomic feature extraction are:

- Identification of LOCUS features: Extraction of identifying information and metadata for each LOCUS. Five lists were created for the characteristics of each locus: (genes, CDS, tRNAs, rRNAs, ncRNAs). These lists were transformed into comma-separated values (CSV) files, serving as intermediate products.
- Unification: For each field of each locus, a column was created in the CSV file. Given variations between source data files, it was necessary to examine and list their columns to unify the extracted information into a single feature file per GBFF file.

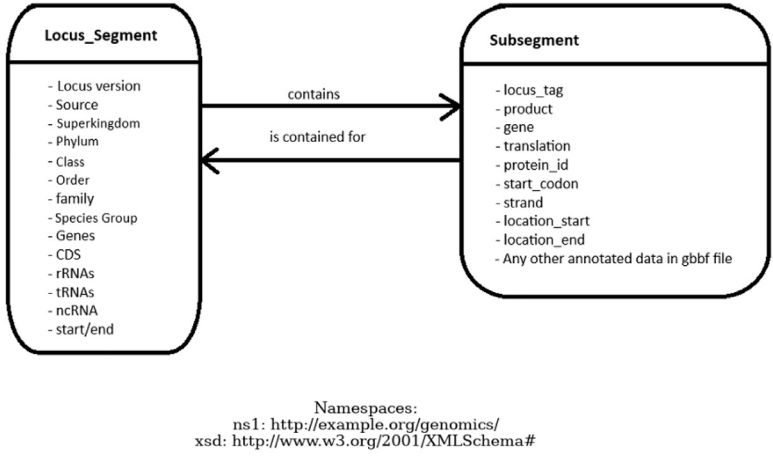
During parsing, only fields in the feature section starting with the standard backslash (\) qualifier format are extracted. Entries deviating from this convention are disregarded to ensure semantic validity in the RDF output, thereby preventing malformed triples.

Step 2. Knowledge Graph. The knowledge graph is composed of vertices corresponding to the loci, encoding taxonomic information as attributes (i.e., knowledge objects). For smaller DNA segments (functional groups), vertices were created to represent them, with attributes detailing the chromosome or plasmid they belong to, and all extracted descriptive information.

The process for constructing the graph is described below:

1. Create individual RDF-Turtle graphs. For each CSV file in the folder:
   - Read the CSV file
   - Create an empty RDF graph
   - Extract data from the CSV and map it to RDF vertices
   - Save the RDF graph as a ".ttl" file
2. Merge individual graphs into a general graph. For each ".ttl" file in the folder:
   - Read the ".ttl" file
   - Add the vertices read into the unified graph

Fig. 3. Sequential transformation of genomic data: a) GBFF file, b) Processed txt file, c) CSV file with extracted features, d) RDF graph in turtle format.



Namespaces:
ns1: http://example.org/genomics/
xsd: http://www.w3.org/2001/XMLSchema#

Note: Only representative properties are shown for clarity, the RDF graph may include any additional annotated data extracted from the original gbff file.

Fig. 4. RDF Schema Representing Classes and their properties.

The result is a ".ttl" file corresponding to RDF syntax, suitable for computational processing while remaining human-readable (Fig. 3.d).

Fig. 4 presents a simplified RDF schema depicting the two main classes (Locus_Segment and Subsegment) along with representative predicates derived from the original genomic features annotations.

All scripts were executed locally using Python 3.8.11 with no use of explicit parallelization libraries, on a workstation with an AMD Ryzen 5 5600 G processor (6 cores, 2.9 GHz) and 32 GB RAM, running Windows 11.

Step 3. Analysis and Querying. The SPARQL Protocol and RDF Query Language are designed for querying RDF graphs. Although SPARQL has a syntax structure, such as SQL, it is specifically tailored for graph queries.

SPARQL allows the execution of pattern-string searches, the filtering of data to retrieve entries meeting specific conditions, and the traversal of relationships, enabling exploration of connections between vertices.

To perform SPARQL queries on the constructed graph, Apache Jena Fuseki (hereafter referred to simply as Fuseki) was used. Fuseki offers several advantages, such as the ability to manage graphs containing thousands of vertices, query optimization, and integration with other tools. Some examples using SPARQL are described below.

Example 1. To retrieve all vertices of type *locus_segment* (genomes) contained in the graph that possesses a gene named "ssrA", the following query can be used, which returns 26 results in 0.011 s. (Table 2):

```
PREFIX ex: ⟨http://example.org/genomics/⟩
    SELECT DISTINCT ?locus_segment
    WHERE {
      ?subsegment a ex:Subsegment;
        ex:parent_segment ?locus_segment;
        ex:gene ?gene.
      FILTER(?gene = "ssrA")
    }
```

Although this search process could also be made by directly scanning the GBFF files, it would be highly time-consuming. Even with automation, it would require opening multiple files to search for the desired data. Now, if we need to identify genomes that include not only a single piece of information, but also a set of features, the inefficiency of direct file searches would be even greater. Through the knowledge graph, however, it is possible to perform composite queries efficiently.

Example 2. To identify genomes that contain both the "ssrA" and "gyrB" genes, the following query can be made, from which returns only two genomes in 0.281 s (Table 3):

```
    PREFIX ex: ⟨http://example.org/genomics/⟩
    SELECT DISTINCT ?locus_segment
    WHERE {
      # Subsegment with gene = "ssrA"
      ?subsegment1 a ex:Subsegment;
          ex:parent_segment ?locus_segment;
          ex:gene ?gene1.
    FILTER(?gene1 = "ssrA")

    # Subsegment with gene = "gyrB"
    ?subsegment2 a ex:Subsegment;
      ex:parent_segment ?locus_segment;
      ex:gene ?gene2 .
    FILTER(?gene2 = "gyrB")
}
```

Example 3. If, in addition to identify the genomes containing both genes, it is also necessary to retrieve specific loci corresponding to each gene, the following query can be executed which returns the results shown in Table 4, within 0.272 s:

```
    PREFIX ex: ⟨http://example.org/genomics/⟩
    SELECT DISTINCT ?locus_segment ?subsegment_ssrA ?subsegment_gyrB
    WHERE {
      # Subsegment with gene = "ssrA"
      ?subsegment_ssrA a ex:Subsegment;
        ex:parent_segment ?locus_segment;
        ex:gene ?gene1.
```

Table 3
Genomes in the graph which contain gene "ssrA".

Locus_segment	Locus_segment
http://example.org/genomics/Locus_Segment_CP016638	http://example.org/genomics/Locus_Segment_CP018405
http://example.org/genomics/Locus_Segment_CP016442	http://example.org/genomics/Locus_Segment_CP018406
http://example.org/genomics/Locus_Segment_CP018399	http://example.org/genomics/Locus_Segment_CP018408
http://example.org/genomics/Locus_Segment_CP017052	http://example.org/genomics/Locus_Segment_CP018410
http://example.org/genomics/Locus_Segment_CP017050	http://example.org/genomics/Locus_Segment_CP018413
http://example.org/genomics/Locus_Segment_CP017048	http://example.org/genomics/Locus_Segment_CP018416
http://example.org/genomics/Locus_Segment_CP018054	http://example.org/genomics/Locus_Segment_CP000440
http://example.org/genomics/Locus_Segment_CP018373	http://example.org/genomics/Locus_Segment_CP000151
http://example.org/genomics/Locus_Segment_CP018418	http://example.org/genomics/Locus_Segment_CP000010
http://example.org/genomics/Locus_Segment_CP018380	http://example.org/genomics/Locus_Segment_NKFA01000003
http://example.org/genomics/Locus_Segment_CP018383	http://example.org/genomics/Locus_Segment_CP012041
http://example.org/genomics/Locus_Segment_CP018389	http://example.org/genomics/Locus_Segment_MDEQ02000019
http://example.org/genomics/Locus_Segment_CP018391	http://example.org/genomics/Locus_Segment_CP018403

Table 4
Genomes in the graph which contain genes "ssrA" and "gyrB".

locus_segment
http://example.org/genomics/Locus_Segment_CP000010
http://example.org/genomics/Locus_Segment_CP012041

Table 5
Locus ID corresponding to genes "ssrA" and "gyrB".

locus_segment	subsegment_ssrA	subsegment_gyrB
http://example.org/genomics/Locus_Segment_CP000010	http://example.org/genomics/Subsegment_BmtmRNA1	http://example.org/genomics/Subsegment_BMA0003
http://example.org/genomics/Locus_Segment_CP012041	http://example.org/genomics/Subsegment_TR70_1468	http://example.org/genomics/Subsegment_TR70_2570

```
FILTER(?gene1 = "ssrA")

# Subsegment with gene = "gyrB"
  ?subsegment_gyrB a ex:Subsegment;
       ex:parent_segment ?locus_segment;
       ex:gene ?gene2.
FILTER(?gene2 = "gyrB")
}
```

It can be observed that among the 200 genomes contained in the graph, only two possess both genes ("ssrA" and "gyrB"). The corresponding loci for these genes were retrieved: BmtmRNA1 and BMA0003 for Locus_Segment_CP000010, and TR70_1468 and TR70_2570 for Locus_Segment_CP012041 (Table 5).

Table 6
Available annotation data of locus "BmtmRNA1".

Predicate	Object
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://example.org/genomics/Subsegment
http://example.org/genomics/Strain	ATCC 23,344
http://example.org/genomics/Reference_DB	taxon:243,160
http://example.org/genomics/locus_End	3510,148.0e0
http://example.org/genomics/locus_Start	1.0e0
http://example.org/genomics/Complementary	Coding
http://example.org/genomics/is_contained_by	http://example.org/genomics/Locus_Segment_CP000010
http://example.org/genomics/Subfeature_End	465,081.0e0
http://example.org/genomics/Subfeature_Start	464,712.0e0
http://example.org/genomics/locus_tag	BmtmRNA1
http://example.org/genomics/Organism	Burkholderia mallei ATCC 23,344
http://example.org/genomics/parent_segment	http://example.org/genomics/Locus_Segment_CP000010
http://example.org/genomics/Molecule_Type	genomic DNA
http://example.org/genomics/Locus_Type	1
http://example.org/genomics/Subfeature_Type	gene
http://example.org/genomics/gene	ssrA
http://example.org/genomics/Type_Material	type strain of Burkholderia mallei

Example 4. To retrieve all available information about a specific locus, for instance BmtmRNA1 (previously identified as the "ssrA" gene in genome CP000010), the following query can be used:

```
PREFIX ex: ⟨http://example.org/genomics/⟩
SELECT DISTINCT ?predicate ?object
WHERE {

ex:Subsegment_BmtmRNA1 ?predicate ?object.
}
```

This query returns detailed information, including the strain, taxonomic ID from NCBI, scientific name, start and end positions within the genomic sequence, and additional data, all retrieved in just 0.008 s (Table 6):

This type of composite query would traditionally require manually scanning the 200 BGFF files used in the construction of the graph to determine the presence of the target genes and extracting the relevant annotation data. Alternatively, a FASTA file could be compiled containing the amino acid sequences of all proteins encoded by the organisms under study and then used to query the amino acid chains of the protein metabolized from the ssrA and ssrB genes via BLAST. This process would return similarity assessments between sequences, and the researcher would be responsible for identifying those corresponding to the genes of interest to determine which organisms possess both. Afterwards, the researcher would still need to return to the corresponding GBFF file to manually extract the annotated information of interest.

The full source code and execution instructions can be found at the Mendeley repository (https://doi.org/10.17632/pt6xn9mgdf.5) in the folder *Scripts and Documentation*, including the requirements to use the SPARQL endpoint through Apache Jena Fuseki.

Limitations

While individual RDF graphs can be generated for larger datasets, the current limit of 200 unified genomes belonging to the *Burkholderia* genus is due to memory constraints encountered during the RDF merging process, the extraction of features and elaboration of individual graphs can be made on much bigger sets. Future work could focus on scalable merging strategies and the use of infrastructure with higher memory capacity to support broader genomic coverage.

Although the number of available genomes exceeds 7000, the subset of 200 genomes was randomly selected, ensuring equal probability of inclusion for each sample and minimizing potential selection bias. Given the expected genomic similarity within members of the same genus, this sample size is enough to demonstrate the technical feasibility and semantic advantages of the proposed RDF-based transformation methodology.

The primary goal of this work is not to construct a taxonomically comprehensive dataset, but rather to provide a functional, reusable proof of concept for knowledge graph elaboration and SPARQL based querying in microbial genomics.

As far as could be determined through the review of GBFF files, every field describing a locus is identified by a backslash symbol (\), followed by the field name and its corresponding value. However, since each locus may contain unique fields if a field does not follow this structure, it cannot be recognized by the parsing process and, consequently, will be excluded from the information extraction and not integrated into the RDF graph.

Although the graph construction was tested with a dataset of 200 genomes, the approach can be scaled according to the number of genomes required in specific studies by different research groups. The methodology employed allows for the creation of new RDF graphs of additional GBFF files that could be incorporated into the SPARQL endpoint, enabling the expansion of the information available for query execution.

The current RDF vocabulary uses a custom namespace for development; future work will include ontology alignment with established genomic ontologies in order to enhance semantic interoperability.

Although this dataset facilitates advanced querying through SPARQL, no formal approach to graph visualization was implemented in the present work. However, it is technically feasible to convert the RDF graph into formats compatible with visualization tools such as Gephi—using Python libraries like rdflib and networkx to export the data to GEXF or GraphML. Nevertheless, integrating such visual representation exceeded the scope of the current study. Future work should include visualization strategies to enhance the interpretability and structural analysis of the graph.

Nevertheless, the constructed graph may serve as a valuable tool for researchers who require performing search for genes, coding sequences (CDS), or specific features within genomes of the *Burkholderia* genus.

Ethics Statement

The authors have read and followed the ethical requirements for publication in Data in Brief and confirm that the present work does not involve human subjects, animal experiments, or any data collected from social media platforms.

CRediT Author Statement

Reynold Osuna-González: Conceptualization, Investigation, Software, Writing - original draft; **Guillermo De Ita Luna:** Supervision, Writing – review & editing; **Rosa María Valdovinos Rosas:** Supervision, Writing – review & editing; **Yagul Pedraza Pérez:** Conceptualization, Writing – review & editing.

Acknowledgements

The author acknowledges the support of the Secretariat of Science, Humanities, Technology and Innovation (SECIHTI) for the doctoral scholarship provided (scholarship number 83,359).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

Burkholderia Genomic RDF Graph (Original data) (Mendeley Data).

References

[1] J.B. Byrd, A.C. Greene, D.V. Prasad, X. Jiang, C.S. Greene, Responsible, practical genomic data sharing that accelerates research, Nat. Rev. Genet. 21 (10) (2020) 615–629, doi:10.1038/s41576-020-0257-5.
[2] B. Louie, R. Higdon, E. Kolker, A statistical model of protein sequence similarity and function similarity reveals overly-specific function predictions, PLoS ONE 4 (10) (2009) e7546, doi:10.1371/journal.pone.0007546.
[3] S.F. Altschup, W. Gish, W. Miller, E.W. Myers, D.J Lipman, Basic local alignment search tool, J. Mol. Biol. 215 (1990) 403–410.
[4] S. McGinnis, T.L Madden, BLAST: at the core of a powerful and diverse set of sequence analysis tools, Nucleic Acids Res. (2004) W20–W25 32(Web Server), doi:10.1093/nar/gkh435.
[5] G.M. Boratyn, C. Camacho, P.S. Cooper, G. Coulouris, A. Fong, N. Ma, T.L. Madden, W.T. Matten, S.D. McGinnis, Y. Merezhuk, Y. Raytselis, E.W. Sayers, T. Tao, J. Ye, I. Zaretskaya, BLAST: a more efficient report with usability improvements, Nucleic Acids Res. 41 (W1) (2013) W29–W33, doi:10.1093/nar/gkt282.
[6] 1.1 RDF, syntax abstract. https://www.w3.org/TR/rdf11-concepts/.
[7] *Annotation file formats*. (2025). https://www.ncbi.nlm.nih.gov/datasets/docs/v2/reference-docs/file-formats/annotation-files/.
[8] Osuna González, Reynold; De Ita Luna, Guillermo; Valdovinos Rosas, Rosa María (2025), "Burkholderia genomic RDF Graph", Mendeley Data, V2, doi: https://doi.org/10.17632/pt6xn9mgdf.6.
[9] National Center for Biotechnology Information (NCBI)[Internet]. Bethesda (MD): National Library of Medicine (US), National Center for Biotechnology Information; [1988] – [cited 2025]. Available from: https://www.ncbi.nlm.nih.gov/National center for biotechnology information. (n.d.). https://www.ncbi.nlm.nih.gov/.
[10] International nucleotide sequence database collaboration. Version 11.3. https://www.insdc.org/submitting-standards/feature-table/.
[11] Locus. (2025). Genome.Gov. https://www.genome.gov/genetics-glossary/Locus.