REVISTA PERSPECTIVAS

VOL. 8, N°1 / ENERO - JUNIO 2026 / e-ISSN: 266-6688
DOI: https://doi.org/10.47187/perspectivas.8.1.250

CONSTRUCCIÓN DE GRAFOS DE CONOCIMIENTO PARA ACELERAR CONSULTAS DE INFORMACIÓN GENÓMICA

Building knowledge graphs to speed up queries on genomic information

Reynold Osuna González	reynold.osuna@viep.com.mx
Guillermo De Ita Luna	deitaluna63@gmail.com

Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla.
Puebla, Pue., México.

RESUMEN

El presente artículo ofrece una visión general de los principales formatos utilizados para almacenar información genómica, así como del proceso para construir grafos de conocimiento a partir de dichos datos. La información extraída de datos del repositorio del National Center for Biotechnology Information es procesada y estructurada en forma de grafos de conocimiento, lo que permite realizar consultas complejas mediante el lenguaje SPARQL. Representar la información genómica en un formato adecuado para la consulta semántica y la inferencia facilita la identificación de relaciones complejas entre genes, proteínas y metabolitos. Esto, a su vez, respalda el descubrimiento de nuevos compuestos bioactivos y promueve avances en campos como la medicina y la biotecnología. Finalmente, se demuestra cómo es posible aplicar consultas simples y compuestas en SPARQL para recuperar relaciones explícitas e inferidas entre las tríadas que conforman los grafos de conocimiento, mejorando así la profundidad y eficiencia del análisis genómico.

Palabras Clave: Información genómica, Grafos de conocimiento, Lenguaje de consultas SPARQL, Bioinformática.

ABSTRACT

This article provides an overview of the main formats used to store genomic information, as well as the process of building knowledge graphs from such data. The information extracted from data in the repository of the National Center for Biotechnology Information is processed and structured in the form of knowledge graphs, allowing complex queries to be performed using the SPARQL language. Representing genomic information in a format suitable for semantic querying and inference facilitates the identification of complex relationships between genes, proteins, and metabolites. This, in turn, supports the discovery of new bioactive compounds and promotes advances in fields such as medicine and biotechnology. Finally, it is demonstrated how both simple and compound queries in SPARQL can be applied to retrieve explicit and inferred relationships among the triples that make up the knowledge graphs, thus improving the depth and efficiency of genomic analysis.

Keywords: Genomic information, Knowledge graphs, SPARQL query language, Bioinformatics.

Fecha de Recepción: 26/11/2025. **Fecha de Aceptación:** 02/01/2026. **Fecha de Publicación:** 28/01/2026.

I. Introducción

La bioinformática utiliza técnicas computacionales e Inteligencia Artificial (IA) para procesar y analizar grandes cantidades de datos biológicos, tales como secuencias de ADN y proteínas en organismos [1]. Esto ha hecho posible la secuenciación de genomas completos, el descifrado de secuencias de nucleótidos y la determinación de la composición génica.

Una de las aplicaciones de la bioinformática es la identificación de metabolitos (cualquier sustancia formada por reacciones metabólicas dentro de un organismo vivo) que puede llevar al descubrimiento de nuevos fármacos o actuar como agentes de control de plagas. Una vez que se ha determinado la utilidad de un metabolito, el siguiente paso es identificar los genes involucrados en su metabolismo, lo que permite la búsqueda de otros organismos capaces de producirlo.

La comparación de cadenas de genes o proteínas entre organismos del mismo género con genomas conocidos es una práctica común. Si existe una coincidencia exacta, se asume la capacidad de metabolizar el compuesto. Sin embargo, debe considerarse la variabilidad genética, ya que pequeñas variaciones en los genes pueden no afectar su función, y un organismo puede producir el metabolito deseado incluso con un subconjunto de los genes identificados.

En 2012, Google anunció que utilizaría Grafos de Conocimiento (Knowledge Graphs o KGs) en sus sistemas de búsqueda, atrayendo la atención de otras empresas tecnológicas [3]. Los KGs representan el conocimiento de una manera computacionalmente viable, con aplicaciones en bioinformática para analizar relaciones biológicas complejas. Los KGs funcionan como bases de datos no relacionales, lo que los hace adecuados para almacenar datos no estructurados.

Sin embargo, la adopción de grafos de conocimiento para representar información genómica también requiere un sistema de consultas. Se demuestra que SPARQL (SPARQL Protocol and RDF Query Language) es un sistema de consultas efectivo para grafos de conocimiento que codifican información genómica.

BLAST (Herramienta de Búsqueda de Alineamiento Local Básico) [2] es una herramienta utilizada para comparar cadenas de genes, buscando automáticamente similitudes funcionales entre secuencias de cadenas genómicas, permitiendo inferir la función de secuencias desconocidas. Se usa ampliamente en bioinformática para comparar secuencias de ADN, ARN o proteínas en bases de datos biológicas para encontrar regiones similares.

Aplicar BLAST a secuencias de interés en un gran conjunto de genomas, almacenados en formato FASTA, es una tarea que requiere mucho tiempo y evalúa coincidencias basadas en similitud. Esta información ayuda a los investigadores a seleccionar organismos con mayor probabilidad de metabolizar el compuesto de interés, confirmando eventualmente la producción de metabolitos en el laboratorio.

BLAST incluye una fase de evaluación, donde se calculan puntuaciones para regiones de similitud basadas en coincidencias exactas, similitudes y diferencias entre las bases alineadas o aminoácidos. Utilizando estadística, se determina la significancia de las similitudes, y los resultados se filtran utilizando umbrales de significancia para reducir falsos positivos.

A diferencia de otros KGs biomédicos a gran escala, nuestro sistema implementa un pipeline automatizado para procesar Archivos Genbank Flat Files (GBFF), permitiendo la generación rápida y reproducible de grafos RDF (Resource Description Framework). Un enfoque similar es adoptado por el marco Petagraph[4], que integra datos multi-ómicos en un KG biomédico unificado, aprovechando más de 180 ontologías y estándares. Petagraph destaca el potencial de los KGs para permitir consultas y análisis eficientes, paralelo a los objetivos de este estudio en el dominio genómico.

II. Metodología

A. Recopilación de Información Genómica

La secuenciación de ADN de diversos organismos es realizada diariamente por biólogos y biotecnólogos, expandiendo rápidamente nuestro conocimiento genético. El objetivo de la secuenciación de ADN es mapear los genes de un organismo e identificar la función específica de cada gen en la producción de proteínas.

Se utilizó el repositorio del Centro Nacional para Información de Biotecnología (NCBI), que contiene archivos genómicos en formato GenBank Flat File (gbff). Estos archivos incluyen información de secuencias de nucleótidos y metadatos, basados en la Tabla de Definición de Características DDBJ/ENA/GenBank de la International Nucleotide Sequence Database Collaboration (https://www.insdc.org/submitting-standards/feature-table/).

El formato gbff categoriza subsecciones de ADN en tres grupos:

1. **Identificador de características (Feature key):** el grupo funcional al que puede pertenecer una subsecuencia.
2. **Ubicación:** la posición de la característica dentro de la secuencia, medida en bases.
3. **Descriptores (Qualifierss):** información adicional que describe la característica.

```
LOCUS       CP011503              3584103 bp    DNA     circular BCT 18-AUG-201
DEFINITION  Burkholderia pyrrocinia strain DSM 10685 chromosome 1, complete
            sequence.
ACCESSION   CP011503
VERSION     CP011503.1
DBLINK      BioProject: PRJNA283474
            BioSample: SAMN03651233
KEYWORDS    .
SOURCE      Burkholderia pyrrocinia
  ORGANISM  Burkholderia pyrrocinia
            Bacteria; Pseudomonadota; Betaproteobacteria; Burkholderiales;
            Burkholderiaceae; Burkholderia; Burkholderia cepacia complex.
REFERENCE   1  (bases 1 to 3584103)
  AUTHORS   Kwak,Y. and Shin,J.H..
  TITLE     Direct Submission
  JOURNAL   Submitted (11-MAY-2015) School of Applied Biosciences, College of
            Agriculture and Life Sciences, Kyungpook National University, 80
            Daehakro, Bukgu, Daegu 702-701, Republic of Korea
COMMENT     Annotation was added by the NCBI Prokaryotic Genome Annotation
            Pipeline (released 2013). Information about the Pipeline can be
            found here: http://www.ncbi.nlm.nih.gov/genome/annotation_prok/

            ##Genome-Assembly-Data-START##
            Assembly Method       :: RS HGAP Assembly Protocol v. 2.0 in SMRT
                                     analysis v. 2.3.0
            Genome Coverage       :: 123.24x
            Sequencing Technology :: PacBio
            ##Genome-Assembly-Data-END##
```

Fig. 1. *Descripción de LOCUS: información general del locus.*

Un archivo GBFF está divido en tres secciones:

- **Descripción de LOCUS:** Locus identifica un sitio físico o ubicación dentro de un genoma (cromosomas, genes, secuencias codificantes o plásmidos). Esta sección también puede incluir la taxonomía del organismo y varios metadatos, así como información sobre la publicación donde la secuenciación fue reportada por primera vez (ver Fig. 1 para un ejemplo).
- **Características (Features):** Contiene información detallada sobre una secuencia funcional de ADN o ARN, como tamaño, tipo de molécula, cepa de microorganismo, fuente de aislamiento, y si es un cromosoma o plásmido (ver Fig. 2 para un ejemplo).
- **Origen (Origin):** Incluye la secuencia de aminoácidos que componen el cromosoma o plásmido descrito, correspondiente al LOCUS (ver Fig. 3 para un ejemplo).

El formato gbff es uno de los más ampliamente utilizados para almacenar cadenas genómicas y aunque a primera vista es un formato estructurado, permite la adición de datos sin formato, como información de texto libre, lo que dificulta la creación de tablas para consultas complejas a través de un sistema de base de datos relacional.

Por lo tanto, nos enfocamos en representar la información genómica en formato RDF (Marco de Descripción de Recursos), que facilita consultas eficientes a través del lenguaje SPARQL.

SPARQL está diseñado para consultar grafos RDF y es capaz de consultar patrones de grafos requeridos y opcionales junto con sus conjunciones y disyunciones. SPARQL también soporta agregación, subconsultas, negación, creación de valores por expresiones, pruebas de valores extensibles y restricción de consultas por grafos RDF fuente [5], reduciendo el tiempo en el que los investigadores pueden determinar organismos candidatos sobre los que experimentar en el laboratorio.

```
FEATURES             Location/Qualifiers
     source          1..3584103
                     /organism="Burkholderia pyrrocinia"
                     /mol_type="genomic DNA"
                     /strain="DSM 10685"
                     /isolation_source="Soil"
                     /culture_collection="DSM:10685"
                     /type_material="type strain of Burkholderia pyrrocinia"
                     /db_xref="taxon:60550"
                     /chromosome="1"
                     /country="Japan"
                     /collection_date="1965"
     gene            22..2484
                     /locus_tag="ABD05_00005"
                     /note="protein-PII uridylyltransferase; disrupted; Derived
                     by automated computational analysis using gene prediction
                     method: Protein Homology."
                     /pseudo
     gene            2795..3514
                     /locus_tag="ABD05_00010"
     CDS             2795..3514
                     /locus_tag="ABD05_00010"
                     /inference="EXISTENCE: similar to AA
                     sequence:RefSeq:WP_019360280.1"
                     /note="Derived by automated computational analysis using
                     gene prediction method: Protein Homology."
                     /codon_start=1
                     /transl_table=11
                     /product="hypothetical protein"
                     /protein_id="AKL98721.1"
                     /translation="MRGRAARVPKAARVRPIATRPPAKARSAASATAVRRPTARPVAA
                     MTTHGPVVQVPKAARARLIATTQRAKGRNAASAIAARRPTARPVAAMTTHGRAAQVPK
                     AAHAHPIATRQPAKGRNAASMTAVRRPTARPVAATTMHVRAAQALTTSSARPIATRLQ
                     AKARSAASATAVRRPTARPAAATTMHVRAAQALTTASAHPIATKPPAKARSAASATAV
                     RRPIARPRRNGRRCTSAPRRR"
```

Fig. 2. *Sección FEATURES: descripción de una secuencia funcional de ADN o RNA.*

```
ORIGIN
        1 ccgacgacgc actgaagcgc gtgggacgat tgcgggctgc cggcgacgct cgcgctcgtc
       61 gccgtcggcg gctacgggcg cggcgagctc gccccctatt ccgacgtcga catcctcgtg
      121 ctgctgccga tgcgcacgac ccggcgctcg atccgcgcat cgaacgcttc atcgggatgg
      181 cgtggatctc ggcctcgaga tcggcagcag cgtgcgcacg gtcgcgcagt gcatcgagga
      241 ggcgtcgcag gacgtcacgg tgcaaacctc gctgctggaa gcgcgccgca tcgtcggcag
      301 caccgcgctg ttcgagcgct tcacggtgcg ctaccacgag gcgctcgacg cccgcgcgtt
      361 cttcacacgc gaaggtgctc gagatgcgcc agcgccacgc gaagttccag gacacgccgt
      421 acagcctcga accgaacgtg aaggaaagcc ccggcgggct gcgcgacctg cagacgatcc
      481 tgtggatcgc gcgtgcggca ggcttcggca gcagctggcg cgagctcgac acgcgcgccc
      541 tcatcaccga tcgcgaagcg cgcgagctgc gccgcaacga aggcttcctg aagacgctgc
      601 gcgcgcggct gcacgtgatc gccggccgcc gccaggacat gctcgtcttc gacctgcaga
      661 cgcaggccgc cgagagcttc ggctaccagc cgacgcaggc caagcgcgcg agcgagcagc
      721 tgatgcgccg ctattactgg ccgcgcgaaa gccgtcacgc agctcgcgac gatcctgatc
      781 cagaacatcg aggcacagct cttccccgcg acgagcggca tcacgcgcgt gctgtcgccc
      841 aatcgcttcg tcgagaagca ggggatgctc gagatcgtcg acgacggcgt gttcgaacgc
      901 catcccgatg cgatcctcga agcgttcttg ctgtacgaaa cgacccgcgg cgtgaagggc
      961 ctgtccgcac gcacgctgcg cgcgctgtac aactcgcgcg aaatcatgaa caacgcgtgg
     1021 cgccgcgatc cgcagaaccg cgacacgttc atgcggatcc tgcagcagcc cgaaggaatc
     1081 acgcacgcgt tccggctgat gaaccagacg agcgtgctcg gccgctacct gctgaacttc
     1141 cgccgcatcg tcggccagat gcaggcacga cctgtaccac gtgtacacgg tcgaccagca
     1201 catcctgatg gtgttgcgcg caacatccgc cgcttcgcgg tcgccgagca tgcgcacgaa
     1261 tacccgttct gcagccagtt gatcggcaac ttcgagcgcc cgtgggtgct gtatgtcgcg
     1321 gcgctgtccc acgacatcgc gaagggccgc ggcggcgacc actcgacgct cggatggccg
     1381 acgcgcgacg cttctgcgcg aacacggaat cgccggcgac gacgcggcgc tgatcgtgtg
     1441 gcctcgtcca gcatcacctg acgatgagcc aggtcgcgca gaagcaggac acgagcgacc
     1501 cgaagtcatc aaagcgcttc gccgaactcg tcggcaacga acggcgcctc aaccagcgct
     1561 ctacctttctg accgtcgccg atatccgcgg cacgagcccg aaggtgtgga aacacgtgga
     1621 agggcaagct gctcgaggat ctgtaccgca tcacgctcgc ggtgctcggc ggcgcgaacc
     1681 ccgatgcaca ctccgagttg aagtcgcggc aggaacaggc gctcgcgctg ctgcgcctcg
     1741 agaccgtgcc cgacgacgcg caccgcgcgc gctgtgggat caactcgacg tcgggcttct
     1801 tcctgcgtca cgatgcggcc cgacatcgca tggcagacac gtgtgctgta ccggcacgtg
     1861 aacgccgaaa ccgcgatcgt ccgcgcgcgg ccgtcgccga tcggcgacgc gctgcaggtg
     1921 ctcgtgtacg tgaaggatcg ccccgacctg ttcgcggggc atctgcgcgt atttcgaccg
```

Fig. 3. *Sección ORIGIN: la secuencia de aminoácidos del LOCUS.*

B. Procesamiento de Información Genómica

La extracción de información genómica del repositorio del NCBI requiere diferentes tareas:

1. **Preprocesamiento:** Limpieza de archivos eliminando saltos de línea innecesarios para consolidar la información del archivo en un formato más uniforme que facilite la extracción de datos.
2. **Identificación de Campos:** Identificación de nombres de campos clave como gene, locus_tag, translation y extracción de sus valores, siempre asociados a la sección a la que pertenecen.
3. **Recopilación de Datos:** Se extrajo la información considerada relevante, incluyendo:

    - Identificadores de secuencias (locus_tag)
    - Posiciones de genes y CDS
    - Secuencias de aminoácidos (translation)
    - Anotaciones funcionales y metadatos adicionales

Se desarrollaron rutinas en Python para automatizar el proceso de extracción. Estas rutinas convierten los datos en texto plano, eliminan caracteres innecesarios y los organizan. Se extrae el LOCUS y sus metadatos, y se crean cinco listas para capturar las características de cada locus, incluyendo genes, CDS, tRNA, rRNA y ncRNA.

Dado que no todos los archivos contienen los mismos campos, las rutinas identifican los campos específicos para cada característica. El resultado son cinco listas, cada una compartiendo una columna que describe el organismo y la sección LOCUS, seguida de columnas adicionales específicas de la fuente de datos. Estas listas se convierten en archivos CSV.

El contenido de las columnas de todos los archivos CSV obtenidos se examina para unificar los datos extraídos correspondientes a cada locus de los diversos microorganismos en un archivo de características único para cada archivo GBFF. Al final, se obtiene un archivo para cada tipo de locus, incluyendo la información de todos los organismos utilizados. A partir de estos datos, se utiliza RDFLib para generar archivos de grafos RDF que representan la información genómica extraída.

El diagrama de flujo de la Fig. 4 se proporciona para dar una mejor comprensión de los pasos seguidos para crear el KG, desde la recopilación de datos hasta la construcción del grafo RDF.

Fig. 1. *Descripción de LOCUS: información general del locus.*

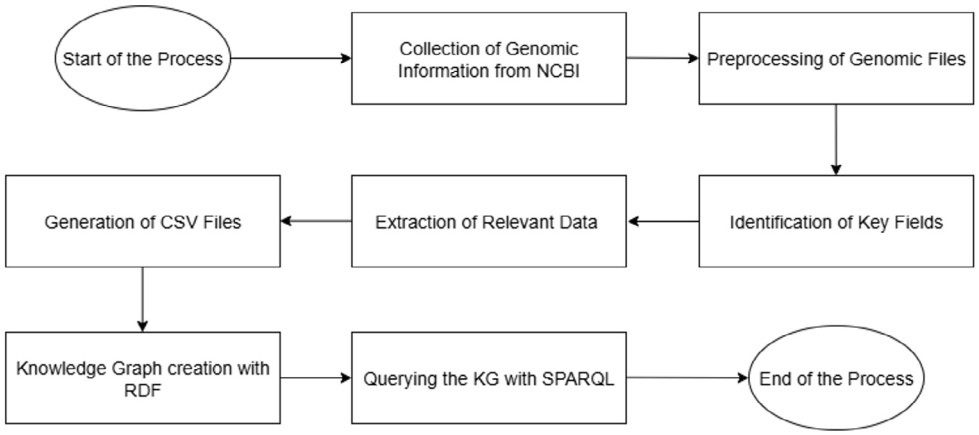

Fig. 4. *Diagrama de Flujo para la recolección de información y construcción del grafo RDF.*

C. Grafos de Conocimiento

Los Grafos de Conocimiento o KG (véase Fig. 4) se consideran una herramienta poderosa para codificar conocimiento. Un grafo de conocimiento se denota como:

$$KG = < E, R, T > \quad (1)$$

Donde E es un conjunto de entidades, T es el conjunto de cola (también es parte de las entidades), y R es el conjunto de relaciones. R representa el conjunto de relaciones y los bordes en R conectan dos nodos para formar una tríada (h, r, t) [6].

En una tríada (h, r, t), existe una direccionalidad implícita de la relación, que comienza desde la entidad "cabeza" hacia la entidad "cola", de modo que puede deducirse que una característica de un grafo de conocimiento es que es un grafo dirigido.

Siguiendo esta definición del KG basada en tríadas, se puede definir el razonamiento sobre grafos de conocimiento como el proceso por el cual, siguiendo una ruta relacional P, se genera una tríada *(h, r, t)* tal que $h \in E, r \in R, t \in T$, pero $(h, r, t) \notin KG$. Las aplicaciones de KG son vastas, desde generar nuevo conocimiento hasta respaldar decisiones.

Se pueden aplicar técnicas de aprendizaje automático para derivar nuevo conocimiento de grafos de conocimiento (KGs). Los sistemas de consulta relacional, por ejemplo, pueden organizar grafos de conocimiento como tablas, habilitando procesos de bioinformática como la búsqueda de metabolitos en secuencias genómicas.

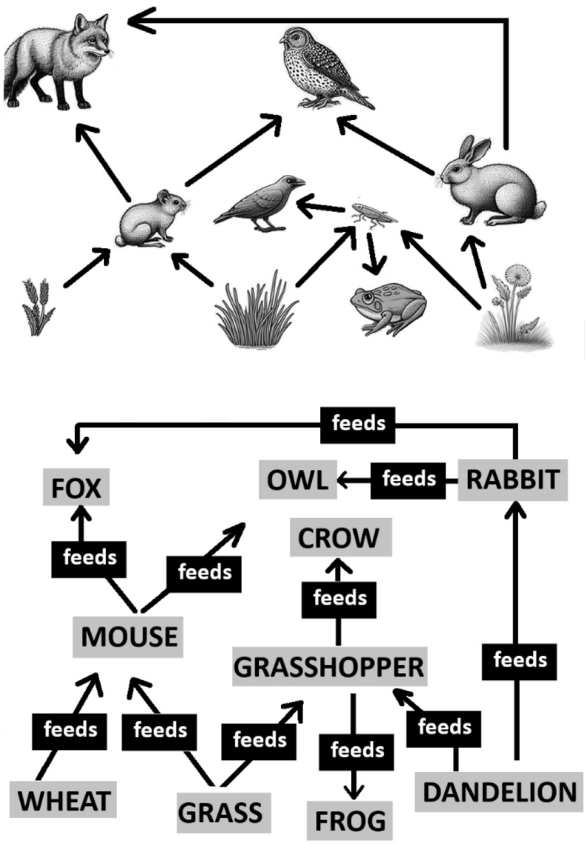

Fig. 5. *Cadena trófica y su correspondiente Grafo de Conocimiento.*

Otro proyecto, *Construcción Escalable de Grafos de Conocimiento e Inferencia en Variantes del Genoma Humano*, se enfocó en representar datos a nivel de variantes de secuencias de ARN de pacientes con COVID-19 en un grafo de conocimiento unificado. Este estudio demostró la utilidad de los grafos de conocimiento en el análisis de datos genómicos, particularmente en la comprensión de relaciones genéticas complejas [8].

Segmentos de ADN (loci, plural de locus) como son cromosomas o plásmidos, conforman nodos (tríadas) del grafo de conocimiento, convirtiendo datos taxonómicos en atributos. Loci más pequeños de ADN funcional tales como genes, secuencias de codificación (CDS) o ARN son nodos vinculados a su cromosoma o plásmido correspondiente, con la información de archivos .gbff como sus atributos (véase Fig. 5). Utilizando archivos CSV como entrada, se crean grafos RDF, con una tríada para cada genoma procesado, como se muestra en la Fig. 6.

Una rutina adicional en Python es responsable de fusionar cada tríada formada a partir del archivo CSV en un único grafo RDF, lo que es útil tanto para crear un nuevo grafo desde cero como para actualizar uno existente con nuevos genomas. El formato RDF facilita la búsqueda de información específica ya que está codificada como una tabla de base de datos.

El resultado de este proceso es un archivo en formato ttl, correspondiente a una sintaxis RDF adecuada para procesamiento por medios computacionales mientras que al mismo tiempo presenta un formato legible por humanos.

III. Resultados

A. Consultas Basadas en Grafos de Conocimiento

Un grafo RDF es una representación de datos estructurada donde la información está organizada en tríadas (sujeto, predicado, objeto). Estas tríadas modelan relaciones entre entidades, haciendo de los grafos RDF una herramienta poderosa para reconocer datos interconectados y realizar consultas complejas.

Desde una perspectiva práctica, un grafo RDF funciona como una base de datos no relacional que permite almacenar y recuperar datos de manera flexible. Los datos forman una estructura de red, facilitando la representación de relaciones jerárquicas, taxonómicas y múltiples conexiones de nodos.

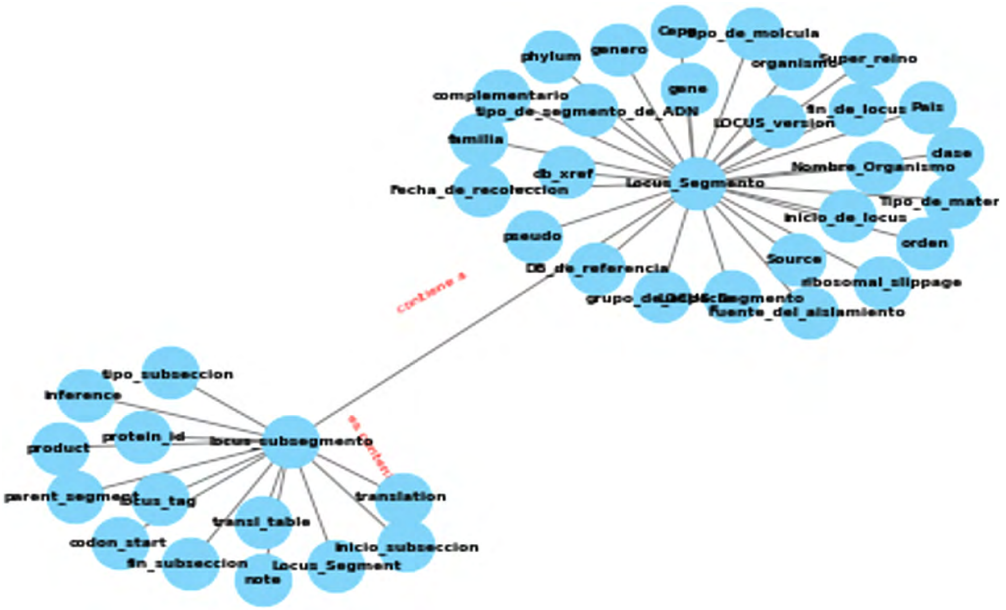

Fig. 6. *Topología del grafo RDF resultante*

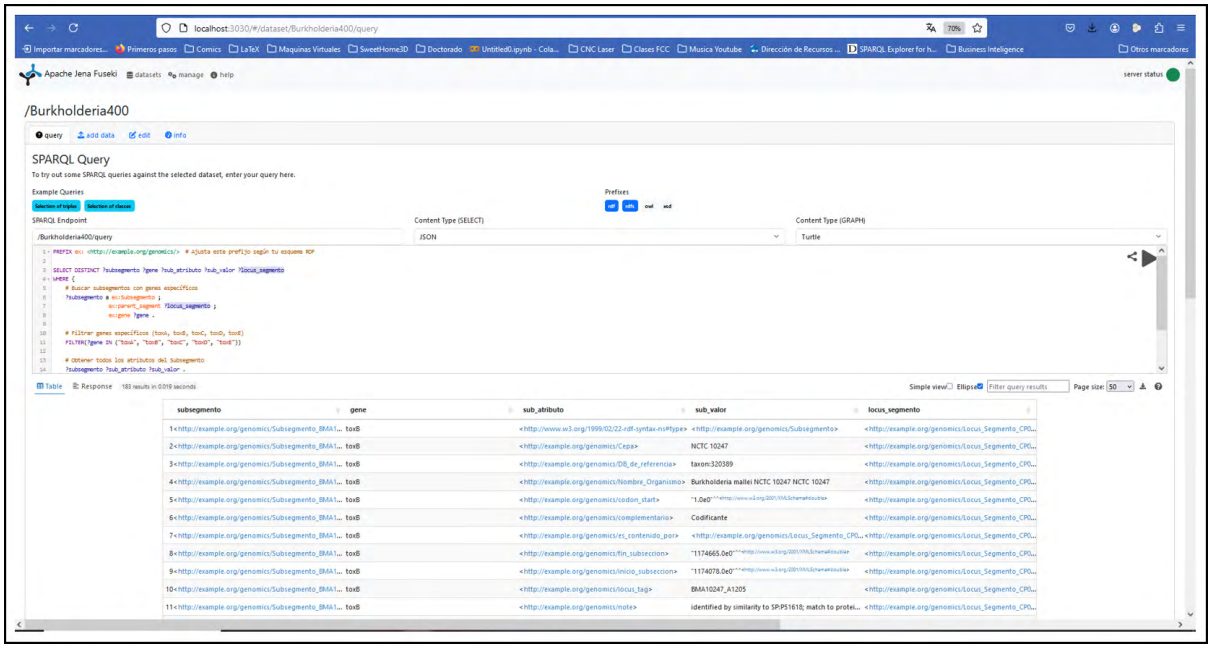

Fig. 7. *Consulta SPARQL en Apache Jena Fuseki de genes involucrados en la metabolización de la toxoflavina.*

SPARQL (Protocolo SPARQL y Lenguaje de Consulta RDF) está diseñado para consultar grafos RDF. Aunque comparte similitudes con SQL, está específicamente diseñado para consultas de grafos de conocimiento. SPARQL permite búsquedas de patrones, filtrado de datos y navegación de relaciones, permitiendo la exploración de conexiones entre nodos. La integración de SPARQL con grafos de conocimiento mejora el análisis de datos genómicos aprovechando ontologías para crear representaciones estructuradas. Esta estructura respalda consultas sofisticadas, enriqueciendo conjuntos de datos existentes [9].

Para consultar un grafo de conocimiento con SPARQL, se necesita un punto de acceso, que sirve como una interfaz para enviar consultas y recibir resultados. Un punto de acceso popular y fácil de implementar es Apache Jena Fuseki (Fuseki), un servidor RDF que importa grafos RDF y permite consultas, devolviendo resultados en varios formatos. Una ventaja clave de Fuseki es su capacidad para manejar grafos grandes con optimización de consultas e integración con otras herramientas.

Utilizando Fuseki, el archivo generado se importó en formato ttl y se realizaron consultas para validar que la información recuperada correspondía a la originalmente contenida en los archivos .gbff.

Por ejemplo, si queremos saber todos los nodos del tipo locus_segment (Genoma) que contienen un gen llamado "ssrA", se realizaría la siguiente consulta:

*SELECT DISTINCT ?locus_segment *
WHERE {
*?subsegmento a ex:Subsegmento ; *
*ex:parent_segment ?locus_segmento ; *
*ex:gene ?gene . *
*FILTER(?gene = "ssrA") *
}

De acuerdo con los datos recuperados, la consulta arroja 26 coincidencias en un tiempo de 0.011 segundos (listados en la Tabla I). Para replicar la misma consulta usando BLAST, el primer paso es descargar la secuencia del gen ssrA en formato FASTA para usarla como consulta. Posteriormente es necesario extraer la secuencia de nucleótidos del genoma anotado gbff y construir una base de datos BLAST local y ejecutar BLASTN (la versión de BLAST para consultas de nucleótidos). A continuación, el investigador debe identificar en los resultados las coincidencias con valores altos de identidad y alineamiento. Finalmente, para

cada ID de secuencia coincidente, se consulta manualmente el archivo gbff correspondiente para recuperar las anotaciones de genes asociadas.

Tabla. I. *Nodos Locus_Segment que contienen el gen ssrA*

Locus Segment Predicate
http://example.org/genomics/Locus_Segmento_CP016638
http://example.org/genomics/Locus_Segmento_CP016442
http://example.org/genomics/Locus_Segmento_MDEQ02000019
http://example.org/genomics/Locus_Segmento_CP017052
http://example.org/genomics/Locus_Segmento_CP017050
http://example.org/genomics/Locus_Segmento_CP017048
http://example.org/genomics/Locus_Segmento_CP018054
http://example.org/genomics/Locus_Segmento_CP018373
http://example.org/genomics/Locus_Segmento_CP018380
http://example.org/genomics/Locus_Segmento_CP018383
http://example.org/genomics/Locus_Segmento_CP018389
http://example.org/genomics/Locus_Segmento_CP018391
http://example.org/genomics/Locus_Segmento_CP018399 http://example.org/genomics/Locus_Segmento_CP018403
http://example.org/genomics/Locus_Segmento_CP018405
http://example.org/genomics/Locus_Segmento_CP018406
http://example.org/genomics/Locus_Segmento_CP018408
http://example.org/genomics/Locus_Segmento_CP018410
http://example.org/genomics/Locus_Segmento_CP018413
http://example.org/genomics/Locus_Segmento_CP018416
http://example.org/genomics/Locus_Segmento_CP018418
http://example.org/genomics/Locus_Segmento_CP0440
http://example.org/genomics/Locus_Segmento_CP0151
http://example.org/genomics/Locus_Segmento_CP00010
http://example.org/genomics/Locus_Segmento_NKFA01000003
http://example.org/genomics/Locus_Segmento_CP012041

La toxoflavina es un metabolito producido por algunas especies del género Burkholderia. Aunque es una toxina, se estudia por sus posibles aplicaciones en control biológico y la síntesis de nuevos antibióticos. Se han identificado cinco genes involucrados en su metabolismo nombrados toxA, toxB, toxC, toxD, y toxE.

Una búsqueda rápida puede realizarse en el grafo construido para identificar genomas que contienen al menos uno de estos genes y recuperar todos los atributos asociados usando la siguiente consulta SPARQL:

```
PREFIX ex: <http://example.org/genomics/>
SELECT DISTINCT ?subsegmento ?gene ?sub_atributo ?sub_valor ?locus_segmento
WHERE {
  # Search for subsegments with specific genes
  ?subsegmento a ex:Subsegmento ;
  ex:parent_segment ?locus_segmento ;
  ex:gene ?gene .
  # Filter specific genes (toxA, toxB, toxC, toxD, toxE)
  FILTER(?gene IN ("toxA", "toxB", "toxC", "toxD", "toxE"))
  # Retrieve all attributes of the subsegment
  ?subsegmento ?sub_atributo ?sub_valor .
}
LIMIT 500
```

Cómo resultado de la consulta, se recupera la información disponible en los archivos originales de los subsegmentos involucrados en la metabolización de la toxoflavina (ver Fig. 7).

Existen grandes colecciones de preguntas en lenguaje natural escritas por humanos y sus correspondientes consultas SPARQL compiladas sobre grafos de conocimiento de bioinformática federados. Estos recursos ayudan en la comprensión de cómo formular consultas que abarquen múltiples conjuntos de datos, facilitando la recuperación completa de datos en investigaciones genómicas [10].

IV. Conclusiones

Este artículo muestra cómo a partir de la recopilación de información genómica de repositorios públicos, como el repositorio del NCBI, los datos pueden ser organizados y procesados para construir grafos de conocimiento.

Al extraer la información del formato GBFF y ser almacenada en tablas relacionales, se puede posteriormente representar la información genómica con un KG (en formato RDF), especialmente orientado a ser consultado y procesado para recuperar tanto información existente como información inferida, a través de ser procesadas por consultas simples y compuestas en el sistema SPARQL (Protocolo SPARQL y Lenguaje de Consulta RDF).

Tener un formato adecuado para consultar e inferir información genómica ayudará a reconocer relaciones complejas entre genes,

proteínas y metabolitos, así como permite respaldar el descubrimiento de nuevos compuestos biológicamente activos e impulsar avances en varios campos, como la medicina y la biotecnología.

Se ha preparado un repositorio que contiene los scripts, grafos RDF, datos fuente (en formato CSV y Turtle), y consultas de ejemplo desarrolladas durante este estudio, disponible en Mendeley Data [11].

V. Agradecimientos

Los autores agradecen el apoyo de la Secretaría de Ciencia, Humanidades, Tecnología e Innovación, México, por la beca para el estudio de doctorado (Número de beca 83359).

VI. Referencias

[1] C. Notredame y J.-M. Claverie, Bioinformatics for dummies (Second Edition). Wiley Publishing, Inc., 2007.

[2] S. F. Altschul et al., "Basic local alignment search tool," J Mol Biol, vol. 215, n.º 3, pp. 403-410, 1990, doi: 10.1016/S0022-2836(05)80360-2.

[3] A. Hogan et al., "Knowledge Graphs," ACM Comput. Surv., vol. 54, n.º 4, Art. 71, may. 2022, doi: 10.1145/3447772.

[4] B. J. Stear et al., "Petagraph: A large-scale unifying knowledge graph framework for integrating biomolecular and biomedical data," Scientific Data, vol. 11, n.º 1, 2024, doi: 10.1038/s41597-024-04070-w.

[5] World Wide Web Consortium (W3C), "SPARQL 1.1 Query Language." [En línea]. Disponible en: https://www.w3.org/RF/sparql11-query/ (accedido el 9 de febrero de 2025).

[6] X. Chen, S. Jia y Y. Xiang, "A review: Knowledge reasoning over knowledge graph," Expert Systems with Applications, vol. 141, 2020, Art. 112948, doi: 10.1016/j.eswa.2019.112948.

[7] F. Feng et al., "GenomicKB: a knowledge graph for the human genome," Nucleic Acids Research, vol. 51, n.º D1, pp. D950–D956, 6 de enero de 2023, doi: 10.1093/nar/gkac957.

[8] S. Prasanna, D. Rao, E.J. Simões y P. Rao, "Scalable Knowledge Graph Construction and Inference on Human Genome Variants," ArXiv, abs/2312.04423, 2023.

[9] E. Cavalleri et al., "An ontology-based knowledge graph for representing interactions involving RNA molecules," Scientific Data, vol. 11, 2023.

[10] J. Bolleman et al., "A large collection of bioinformatics question-query pairs over federated knowledge graphs: methodology and applications," 1990, doi: https://doi.org/https://doi.org/10.1016/S0022-2836(05)80360-2.

[11] R. Osuna González, G. De Ita Luna, R. M. Valdovinos Rosas y Y. Pedraza-Pérez, “Burkholderia Genomic RDF Graph”, Mendeley Data, V6, 2025, doi: 10.17632/pt6xn9mgdf.6.