BioHackrXiv

Report: Workshop on connecting Knowledge Graphs with BioChatter

Cyril Pommier [1], **Sebastian Beier** [2], **Pedro M Barros** [3], **Johann Confais** [1], **Nicolas Francillonne** [1], **Hugo Rodrigues** [3], **Hiromi Kajiya-Kanegae** [4, 5], **Ryokei Tanaka** [4], **Bruno Costa** [6, 7], **Raphaël Flores** [1], **Célia Michotey** [1], **Helene Rimbert** [8, 9], **Pierre Larmande** [10, 11], **Carissa Bleker** [12], **Maxime Multari** [13], and **Sebastian Lobentanzer** [14, 15]

1 Université Paris-Saclay, INRAE, BioinfOmics, URGI, 78026, Versailles, France **2** Institute of Bio- and Geosciences (IBG-4 Bioinformatics), Bioeconomy Science Center (BioSC), CEPLAS, Forschungszentrum Jülich GmbH, 52425 Jülich, Germany **3** Instituto de Tecnologia Química e Biológica António Xavier, Universidade Nova de Lisboa (ITQB NOVA), Av. da República, Oeiras 2780-157, Portugal **4** National Agriculture and Food Research Organization(NARO), Institute of Crop Science(NICS), Tsukuba, Japan **5** National Agriculture and Food Research Organization(NARO), Research Center for Agricultural Information Technology (RCAIT), Tsukuba, Japan **6** Instituto de Engenharia de Sistemas e Computadores - Investigação e desenvolvimento (INESC-ID), Rua Alves Redol, 1000-29 Lisboa **7** BioISI, Faculdade de Ciências - Universidade de Lisboa (FCUL), Campo Grande, 1749-016 Lisboa **8** National Research Institute for Agriculture, Food and Environment (INRAE), Genetics, Diversity and Ecophysiology of Cereals (GDEC), Clermont-Ferrand, Auvergne, France **9** AuBi - Plateforme Auvergne Bioinformatique (Université Clermont Auvergne, Bât. Turing, Mésocentre Clermont Auvergne, 7, avenue Blaise Pascal, TSA 60026, 63178 Aubière cedex - France **10** University of Montpellier, IRD, CIRAD, DIADE, Montpellier, France **11** South Green Bioinformatics Platform, IRD, CIRAD, INRAE, Bioversity, Montpellier, France **12** National Institute of Biology, Večna pot 121, 1000 Ljubljana, Slovenia **13** INRAE, Université Côte d'Azur, CNRS, Institut Sophia Agrobiotech, Sophia-Antipolis, France **14** Heidelberg University, Faculty of Medicine and Heidelberg University Hospital, Institute for Computational Biomedicine, Heidelberg, Germany **15** Open Targets, European Bioinformatics Institute (EMBL-EBI), Hinxton, Cambridge, United Kingdom

BioHackathon series:
BioHackathon LLM Versailles 2024
Versailles, France, 2024
BioHackrXiv

Submitted: 05 Mar 2025

Background

In the hackathon subject to this document, biologists, computer scientist, and bioinformaticians gathered to lower the barriers of interacting with integrated datasets represented in knowledge graphs (KG) based on data from biological and agricultural sciences. We worked on the creation and utilisation of Knowledge Graphs (KGs) using the BioCypher framework with several use cases that either had already existing KG or that were using identified standards sources such as the BreedingAPI (BrAPI). To propose more biologist-friendly interfaces, we explored the capabilities of current Large Language Model (LLM) technologies using BioChatter, which allows researchers to query KGs and APIs in natural language. As many researchers are not semantic experts this will enable them to find interesting and integrated datasets without being familiar with KGs or query languages such as SPARQL or Cypher. To evaluate our approach we also discussed a set of scientific questions to identify suitable solutions. The project combines success stories from previous BioHackathon Germany, BioHackathon Europe and BioHackathon Japan work. A novelty is the extension of our generic user-friendly knowledge management infrastructure towards the agroscience domain. In the present paper, we present use case per use case what was achieved as well as an outlook based on these results.

BrAPI use case

Objectives

The field of plant research is characterized by vast amounts of data generated from diverse sources and studies, resulting in significant data dispersion and heterogeneity. Researchers often face challenges in accessing, integrating, and analyzing these dispersed datasets due to their low findability and to differences in formats, structures, and standards. Over the past ten years, data standards and API have been built to formalize the plant trait and genetic data, in particular with MIAPPE and BrAPI.

Biologists often require tools that enable them to aggregate information on specific plant genetic resource accession or trait of interest efficiently. To address this demand, we developed a system that allows users to query data in text format and retrieve results from BrAPI in JSON format.

Scientific questions and Outcomes

Considering the available datasets, biologist and breeders users would like to get data to explore the following questions: - What are the trees older than 12 years, in CSV format, with location and related experiment description. - Are there any trials that measured metabolome-related traits? - What are the reference information for the accession "EM00002", plus its cross references including the experiments and datasets (phenotyping, genome sequence, ...). - Provide information on the wheat accession most frequently cultivated in the northernmost wheat trials in a ready to use Jupyter notebook. - Which variety showed the highest shoot dry weight in Clermont-Ferrand? - What datasets allows to study relation between flowering phenology and yield or disease X. - Which grape varieties can reach grape maturity in September? - Which european rice variety has high protein content in their grains?

The work was conducted with BrAPI and MIAPPE datasets. For BrAPI based development, we used POPYOMICS (https://urgi.versailles.inrae.fr/faidare/brapi/v1/trials/dXJuOklOUkFFLVVSR0kvdHJpYWwvMjQ%3D) and Wheat INRAE Cereal Network (https://urgi.versailles.inrae.fr/faidare/brapi/v1/trials/dXJuOklOUkFFLVVSR0kvdHJpYWwvNw%3D%3D). We developed two approaches

Generate BrAPI queries using BioChatter LLM tool binding

The code necessary to fine-tune biochatter is available in the following pull request: https://github.com/biocypher/biochatter/pull/211

The Breeding API allows to get data answering questions such as "What germplasms belong to the Vitis genus and species vinifera?". In that simple scenario, a unique web service call generates a single list of germplasms object. But more complex questions, such as "Which grape varieties can reach grape maturity in September", there is a need to chain multiple call. But this implies to know how to to navigate the Breeding API endpoints using SWAGGER for instance (https://urgi.versailles.inrae.fr/faidare/swagger-ui/index.html#/Breeding%20API/searchGermplasm). This is not necessarily straightforward especially for biologists who need some initial training on this subject. We tried to use BioChatter to avoid this complexituy. Users can simply enter text to automatically generate API queries, execute the queries and output the results. This approach streamlines interactions with APIs by making complex data retrieval processes accessible through natural language input. But creating a chain of BrAPI calls proved beyond the framework capability during this workshop. Indeed, making first a germplasm call to get the list ob biological material then looping and making a subsequent series of call to get corresponding data was too complicated without deeply fine tuning BioChatter. There was a risk to overspecify the LLM approach, preventing generalization and making it difficult to maintain. In such a situation, at the time of wirtting, programatically chaining the API calls seemed more robust.

BrAPI to Knowledge graph with a BioChatter chatbot to generate queries

The BioCypher adapter is available here : https://github.com/gnpis/biocypher-brapi For those less familiar with data, there is a need for tools that can provide text-based answers to text-based queries. Such tools would allow users to use trait information for biological analysis and breeding purposes without requiring expertise in working with APIs or databases, making data more accessible and user-friendly. Therefore, to be able to really do cross datasets and datatype querying, we focused on the capabilities of Knowledge Graphs using BioCypher to build a generic BrAPI data adapter. It works on a set of JSON files extracted from a BrAPI endpoint. We tried several approaches, and the most promising one was to use BioChatter to generate Cypher queries to get data from the knowledge graph. This proved to work well with questions such as "What are the studies that used the variety APACHE". The queries were functional and all the expected data was successfully returned.

Discussions

Regarding the question "Are there any trials that measured metabolome-related traits", a potential solution would be to use the chatbot to discuss with the users to (i) identify traits of interest within the different trait ontologies, (ii) validate this with the user and (iii) use the trait list to get the data from BrAPI or from a KG.

Regarding the retrieval of a specific Biological Material accession there is a challenge related to multiple IDs for related biological material, such as Biological materialID, genetic resources DOI, Accession Number, etc… Correctly using those IDs highly depends on the context and it is not easy to alleviate all ambiguity to know weither the ID namespace is Biosample, accession number, or other. The LLM does not seem to be a way to solve the synonyms and multiple IDs problem, at least for now.

Future Work

For researchers less familiar with JSON, the system will be designed to output well-structured data in CSV or tab-delimited formats or even generate Jupyter notebook files, providing flexible options to facilitate data analysis such as GWAS and genomic prediction. To improve the search results, it is essential to implement a mechanism that refines results through follow-up queries. This approach would enable features such as narrowing down options by presenting candidate traits or addressing synonyms within accession names, thereby facilitating more precise and meaningful responses. For example, even data stored in databases that previously lacked BrAPI compatibility can be made accessible by converting the data into an BrAPI-compatible format. By establishing this framework, it becomes significantly easier to build systems that can respond to text-based queries, streamlining access to and interaction with such data.

RDF adapter use case based on PPEO

Facilitating data reuse and knowledge discovery is increasingly vital in plant science research, yet this is specially challenging in the field of plant phenotyping due to its inherent complexity and heterogeneity. The MIAPPE (Minimum Information About a Plant Phenotyping Experiment) metadata standard is currently a key resource to enable interoperability between plant phenotypic databases. Additionally, the MIAPPE data model has been previously formalised in OWL as the Plant Phenotyping Experiment Ontology (PPEO).

Objectives and scientific questions

The objective of this use case was to enhance the interoperability and utility of PPEO by converting it into a Biocypher knowledge graph, facilitating its integration into modern data analysis and retrieval systems. More specifically, the aim was to enable the structured knowledge from PPEO to serve as a Retrieval-Augmented-Generation (RAG) resource for a Large Language Model (LLM).

Outcomes

We developed an adapter (https://github.com/forestbiotech-lab/RDF_Adapter) that serves as a middleware for transforming the PPEO ontology into a Biocypher-compatible knowledge graph. It extracts classes, data properties, and object properties from a PPEO knowledge graph, using SPARQL queries and converts them into nodes and edges that align with the Biocypher framework.

This integration allows the LLM to access detailed, hierarchical, and semantically enriched data, thereby improving its ability to generate accurate, context-aware responses and perform advanced reasoning tasks informed by domain-specific ontology-driven knowledge.

Future Work

Future work aims to enhance the integration process by automating schema extraction directly from the RDF source. This approach would involve leveraging SPARQL queries to extract not only class and property instances but also the underlying schema structure, including class hierarchies, data property domains and ranges, and object property relationships. By formalizing and dynamically generating a Biocypher-compatible schema from the RDF source, this process would ensure greater scalability and adaptability for various ontologies. It would also reduce manual interventions and provide a more standardized foundation for knowledge graph generation. This automated schema extraction could further enable seamless updates as the source ontology evolves, ensuring the Biocypher graph remains current and aligned. Ultimately, this advancement would streamline the integration of diverse RDF-based ontologies into RAG systems, improving their utility for enhancing LLM capabilities.

GitHub repositories and data repositories

- Adapter https://github.com/forestbiotech-lab/RDF_Adapter
- Dataset https://github.com/forestbiotech-lab/RDF_Adapter/tree/master/sources
- PHENO https://brapi.biodata.pt

Plant translational database : orthologous database with Wheat, Rice and Arabidopsis

With the advance of long read sequencing, more and more high quality wheat genome are now available in public databases such as Ensembl Plants.
Even if high resolution annotations are released along with these genomes, most of the biological knowledge remains in the scope of models species such as *Arabidopsis thaliana* or *Brachypodium distachyon*.

Objectives and scientific questions

The objective of this work group is to integrate data from several species of interest such as Wheat or Rice, to allow the knowledge transfer from model species to help answer question scientists may have.

Outcomes

A BioCypher instance has integrated the below data which is available as a tar.gz file on RechercheDataGouv: https://doi.org/10.57745/5UNPZQ

Gene

- Arabidposis thaliana biomart from Ensembl Plants: `mart_ath_tair10_export.txt`
- Rice (Oryza sativa japonica nipponbare - IRGSP annotation) biomart from Ensembl Plants: `mart_osjaponica_irgsp1_export.txt`
- Wheat Chinese Spring reference sequence: `mart_tritaestivum_CS_IWGSCv1.1_export.txt`
- Wheat Renan cultivar reference sequence: `TaeRenan_refseqv2.1_genesHC.tsv`

Wheat v1.1, *Oryza sativa cv. japonica* and *Arabidopsis thaliana* gene annotations were extracted from EnsemblPlant using BioMART. The details of the extraction can be found on GitHub https://github.com/gnpis/wheatomics-biocypher/blob/dev/wheatomics-data-import/README.md.

GO annotations

Contains annotations from TAIR (`tair_annotations.gaf`) as well as from Oryzabase (`OryzabaseGeneListEn_20241017010108.txt`)

Homology data

A list of Orthologs genes between Wheat (Chinese Spring cultivar), Rice and Arabidopsis thaliana, obtained from Ensembl Plants BioMart portal: `Wheat_othologs_with_arabido_and_O.Sativa.japonic.txt`

As well as a mapping between the structural annotation of Chinese spring (IWGSC RefSeq v2 and version 1) and with Renan cultivar gene list: `TaeRenan_refseqv2.1_CORRESPONDANCE_CSv1_CSv2.txt`

Wheat orthologies with *Oryza sativa cv. japonica* and *Arabidopsis thaliana* were extracted from EnsemblPlant using BioMART. The details of the extraction can be found on GitHub https://github.com/gnpis/wheatomics-biocypher/blob/dev/wheatomics-data-import/README.md.

RNAseq transcriptomics data

For transcript names, a prefix was constructed to schematize the different conditions (ie. `TaeRnG100T.3`):

- Tae: triticum aestivum
- Rn: Renan
- G/L/R/S: **g**rain/**l**eaf/**r**oot/**s**tem
- 100/250/500/700/Z13/Z32/Z61: number of degree-days or Zadok stage(Konzak, 1974)
- C/T: control/stress **T**emperature

The files ***abundance.tsv** contain the gene expression level for the 14 conditions. FPKM and TPM are added at the gene level (sum of values for the different gene isoforms).

The ***refmap** files contain mapping between Renan reference genome, the main isoform and the list of transcripts foubd in the experiment, the first one being used for mapping with the abundance files.

Future Work

A script cleaning needs to be done in order to have a seamless integration process.

BioChatter needs to be used more in depth on this dataset to expose the integrated data to end users and getting feedback on the LLM benefits.

Jupyter notebooks, GitHub repositories and data repositories

- GitHub repository: https://github.com/gnpis/wheatomics-biocypher
- RechercheDataGouv: doi:TODO

Mobile element knowledge graph use case

Transposable elements (TEs) are mobile DNA sequences that propagate within genomes, representing a significant source of genetic variation and molecular innovation. Cis-regulatory elements (CREs) are genomic regions involved in the regulation of gene expression. It has been demonstrated that transposable elements, during their movement through the genome, may carry CRE across the genome, thereby impacting gene transcription. Indeed, it has been hypothesised that ancient transposable elements may have been co-opted by flowering gene regulation networks (Baud Agnès, 2022). To this end, a knowledge graph was constructed, integrating data from Quesneville et al. (Baud Agnès, 2022). The identification of suitable candidates is a challenging endeavour; this knowledge graph was developed to facilitate the identification of such candidates with a maximum of clues. The objective of this knowledge graph is to investigate the scientific question of the relationship between mobile elements, such as transposable elements, CREs and gene regulation networks.

Objectives and scientific questions

The objective of this endeavour is to facilitate the process of querying our knowledge graph for users who are not familiar with the Cypher language. To this end, a Biochatter instance has been integrated with our knowledge graph, with the establishment of an in-house ontology and schema configuration file.

Outcomes

We managed to connect BioChatter to our database successfully. Since we were aiming to use an already existing database, we tried to do this directly without using biocypher. To do that, we needed to harmonize and structure our data model, using a custom ontology to describe our nodes and their links. Based on this ontology, we set up our schema_config.

We first tried simple queries between two nodes in Biochatter's prompt. However, these queries were not consistently rendered correctly by the tool. This was due to the orientation of the relationship between our nodes, which was not clear to Biochatter. To resolve this issue, we added « source » and « target » to the schema_config file.

Another issue we encountered was related to the value of properties. Contrary to the nodes, relationships and properties, we do not have a controled vocabulary (like ontologie) for our properties values. Exact matching of the property value is necessary to request the knowledge graph. For example, "Flowering", "flowering","floraison" have the same meaning but may differ by language, capital letter etc. Our "trait" property has the value "flowering" but cannot be found if the query is not written in the exact same way. Without prior knowledge of property values, querying them seems hasardous. Should properties values also be limited to a controlled vocabulary ?

Future Work

Future work may be the development of biocypher parser to manage the data of our database to ensure a better connectivity with biochatter and a better schema_config.yml.

Stress Knowledge Map use case

Stress Knowledge Map (SKM, (Bleker et al., 2024)) is a publicly available resource containing current knowledge of biochemical, signalling, and regulatory molecular interactions in plants: notably, a highly curated model of plant stress signalling (PSS) containing 751 reactions, implemented as a knowledge graph. PSS was constructed by domain experts through curation of literature and database resources.

While an interactive explorer is available for PSS, the existing interface does not allow for complex queries, and much of the rich metadata is not directly available to users without needing to download and parse exports. Integrating a (RAG-enabled) LLM with PSS will enable researchers to query the knowledge graph using natural language, eliminating the need to parse data files, learn Cypher, or be deeply familiar with the schema details underlying the knowledge graph.

The focus during the workshop was to develop an interface to PSS using BioCypher (PSS-BioCypher) and subsequently develop and test a BioChatter interface.

Outcomes

Developing PSS-BioCypher

Three challenges had to be tackled in developing PSS-BioCyper:

1. PSS is a **real-time database**, in that an online contribution interface interface allows users to add new information based on novel biological knowledge. On average 7 updates are made to PSS per week, while some weeks have seen over 100 updates. Users expect to be able to immediately access new or updated information in the database. For this reason, instead of creating PSS-BioCypher from flat file exports, the PSS adapter was developed to be directly fed from the Neo4j database.

In the future, the aim is to use this as a basis to incrementally build PSS-BioCypher, based on real-time updates to PSS. Feeding directly from the database will also allow updates to the schema and metadata, by only updating the adapter and not having to additionally update flat file exports.

2. PSS takes into account cross species information and genetic redundancy by grouping genes that take part in the same functions into **functional clusters**. To incorporate this information into PSS-BioCypher, two approaches could be taken:
    - Include functional clusters as nodes and as reaction participants. To enable queries on the gene information level (e.g. gene names and gene identifiers), additionally add gene nodes with `gene to functional cluster` edges.
    - Explode each reaction edge on a functional cluster across all the genes assigned to the functional cluster.

To stay closely aligned to the existing PSS schema, the first option was used.

3. PSS has a complex **schema**, based in reactions connection interacting entities (Fig. 1), instead of direct pairwise interactions. While pairwise interactions are more intuitive in a knowledge graph formalism, the richly curated detail of PSS as a model, its cross-species compatibility, and the need to be able to exchange the information in PSS with other domain standards (such as SBML, SBGN) means the schema of PSS is somewhat convoluted for users to interact with.

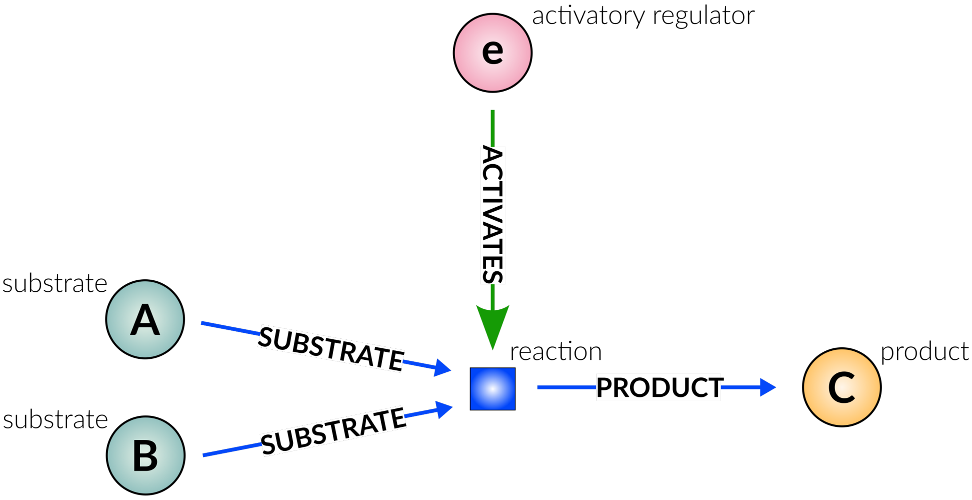

Figure 1: Figure 1: PSS schema of a catalysis reaction.

While the ontology would allow PSS-BioCypher to maintain the reaction based formulation, for the intuition of both the user and ability for the LLM to interpret the structure, the schema of PSS was projected from reactions to pairwise interactions between the reaction participants. As an example, the reaction in Fig. 1 was projected as in Fig. 2.

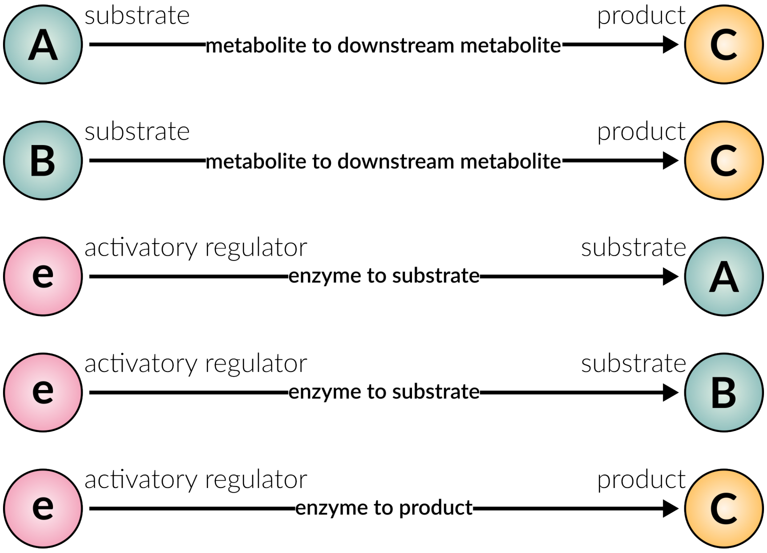

Figure 2: Figure 2: PSS-BioCypher schema of a catalysis reaction.

Including the depicted catalysis, PSS has nine defined reaction types, each of which was projected to the new schema. The `schema_config.yaml` file (in the GitHub repository) contains information on how the projection was done.

Scientific questions

In preparation for the workshop, current users of SKM were surveyed for potential questions about PSS to set to the LLM. The following list of questions was compiled from the responses:

- Which proteins regulate MYC2?
- Which genes are regulated by MYC2?

- Does ABA regulate MYC2, or is it the other way around?
- In which pathways is gene X involved?
- Are genes X, Y, Z regulated by the same transcription factor(s)?
- Is NPR1 important in a plant's response to a bacterial infection?
- What is the most plausible hypothesis for upregulation of genes X, Y, Z and downregulation of B, C, D observed in my RNA-seq experiment where I treated plant leaves with a phytohormone J?

These questions where used in testing the BioChatter interface.

SKM BioChatter

BioChatter was connected to PSS-BioCypher in two manners: by deploying a BioChatter-light instance and by connecting a jupyter notebook. The BioChatter light instance was successfully connected to PSS-BioCypher and could be used to ask simple questions. A notebook (available in the GitHub repository) was used to reproducibly test more questions and interact with the LLM via BioChatter functionalities `BioCypherPromptEngine` and `BioCypherQueryHandler`.

BioChatter was able to provide correct Cypher to answer to a number of simple questions, for example:

> How many genes belong to the MYC2 functional cluster?

```
MATCH (:FunctionalCluster{name: 'MYC2'})-[:GeneBelongsToFunctionalCluster]->(g:Gen
RETURN COUNT(g) as numberOfGenes;
```

> Which functional cluster does the gene with tair identifier AT1G64280 belong to?

```
MATCH (g:Gene {tair: 'AT1G64280'})-[:GeneToFunctionalCluster]->(fc:FunctionalClust
```

It was also successful with more complex questions:

> Which nodes does MYC2 transcriptionally regulate?

```
MATCH (fc1:FunctionalCluster)-[:TranscriptionalActivation|:TranscriptionalInhibiti
WHERE fc1.name = 'MYC2'
RETURN fc2
```

> Which nodes does AT1G32640 transcriptionally regulate?"

```
MATCH (g:Gene {tair: 'AT1G32640'})-[:GeneBelongsToFunctionalCluster]->(fc:Function
RETURN targetGene
```

In the above two cases, the LLM used the target designation ("MYC2" vs "AT1G32640") to correctly deduce the need to anchor the query on a functional cluster vs on a gene. However, in general it was not always successful.

Another issue encountered was inferring the edge type of interest. When not explicitly including the type of interaction in the question, the LLM did not always correctly deduce the difference between physical interactions and functional cluster membership, the latter denoted by `gene belongs to functional cluster` edge (a BioLink `gene to gene family association`). The following question was intended to result in a list of functional clusters that have molecular interactions with MYC2, but instead the provided Cypher results in the functional clusters that the gene MYC2 belongs to:

> Which functional clusters does MYC2 interact with?

```
MATCH (:Gene {name: "MYC2"})-[:GeneBelongsToFunctionalCluster]->(fc:FunctionalClus
RETURN fc.name, fc.description, fc.url;
```

A more explicit question improved the result:

> Which other functional clusters does the MYC2 functional cluster have a molecular interaction with?

```
MATCH (:FunctionalCluster {name: "MYC2"})-[:ProteinProteinInteraction]->(otherClus
RETURN otherCluster.name, otherCluster.description
```

The more complex scientific questions provided by the users will require improvements to the prompt and further development of the PSS-Biocypher scheme. However, the results here are very promising the LLM is able to, at this point, help users generate Cypher queries to interrogate PSS.

Future Work

Guided by the user provided questions and the current performance of the LLM, the BioCypher schema for PSS will undergo a number of iterations, improvements, and refinements. One immediate avenue for improvement is exploring the second option of the proposed two methods to deal with functional clusters, as the LLM occasionally struggled to extrapolate gene function via functions assigned to the functional clusters. Another issue we came across was inconsistent interpretation of gene name vs gene identifier (TAIR). This would also need improvements, perhaps by user instructions.

Complementary to PSS, the Comprehensive Knowledge Network (CKN) contains over 26,234 entities and 488,390 pairwise interactions. CKN was constructed from a combination of database mining and manual curation, and was last updated in 2023. We are currently in the process of incorporating the BioCypher workflow into our planned update to CKN. In this endeavour, we hope to contribute a number of adapters to the BioCypher project (including the PSS-BioCypher adapter). This will also generate a much larger knowledge graph with the same schema as PSS-BioCypher, compatible with BioChatter.

Finally, we plan to integrate a BioChatter RAG assistant in the SKM web page. The aim would be for the assistant to answer questions about the SKM knowledge graphs, provide information about the database schema, and help users formulate queries.

It would be interesting to investigate, if such an assistant could also aid in curation of the database in the form of identifying missing information, providing literature reviews, and summarising new publications in the context of the knowledge currently in SKM.

GitHub repository

The following repository contains the full workflow and instructions to prepare PSS-BioCypher, and deploy a BioChatter-light instance that can provide cypher queries for PSS-BioCypher: * https://github.com/NIB-SI/skm-biocypher

Chem and Plant KG use case

In recent years, the integration of diverse scientific resources has become an increasingly crucial aspect of addressing complex research challenges that span multiple disciplines. A considerable number of valuable datasets pertaining to plant biology, biochemistry, and metabolic pathways, among other areas of the life and natural sciences, remain confined to disparate repositories, thereby limiting their potential for cross-disciplinary utilisation. The integration of disparate resources through the construction of a unified knowledge graph facilitates enhanced data interoperability, thus enabling researchers to discover, access and analyse information in a more efficient manner. Such an approach not only enhances the accessibility of the resources in question, but also drives new insights by linking related datasets and uncovering hidden relationships across domains. However, a limitation of this approach is that it is only accessible to researchers with a high level of expertise in programming and a detailed understanding of SPARQL queries.

This use case is based on the knowledge graph developed during the BioHackathon Europe 2023 event, as detailed in this report (DOI: 10.37044/osf.io/yxunp). The knowledge graph integrates

JSON-LD Bioschemas markup from a variety of resources in the life and natural sciences, with a particular focus on plant sciences and (bio)chemistry research. The data sources contributing to this knowledge graph include COCONUT, a database of natural products; e!DAL-PGP, an endpoint repository of plant research data; MassBank, a repository for mass spectrometry data; MetaNetX, a platform for metabolic network reconstructions; NMRXiv, a repository for nuclear magnetic resonance data; WikiPathways, a community-curated database of biological pathways; and SABIO-RK, a database for biochemical reactions and their kinetic properties.

A GraphDB instance was established to serve as the foundation for the knowledge graph. This database enabled the integration and querying of structured metadata, facilitating efficient navigation across datasets and resources. Implementation was conducted using the Bioschemas framework, which standardised metadata schemas and ensured consistency across the sources.

Objectives and scientific questions

The principal objective of this use case in the context of the workshop was to investigate strategies for utilising existing knowledge graph data and integrating it with large language models, with the intention to improve query and discovery capabilities. Two specific objectives were identified and defined as follows:

1. This approach involved using the available Turtle (.ttl) files from the previous Bio-Hackathon to construct a knowledge graph within the BioCypher framework. BioCypher, which relies on the Neo4j graph database, has been developed with the objective of facilitating the organisation and querying of large-scale biological data. The objective was to ascertain how BioCypher's infrastructure could be employed to establish a connection between the knowledge graph and BioChatter, a tool designed for interactive exploration and querying of data using natural language.
2. The second approach aimed to reduce the time required for developing the knowledge graph itself, instead focusing on directly integrating the data with BioChatter and LLMs. The objective was to generate a Neo4j-based knowledge graph from the exported Turtle files in order to concentrate efforts on connecting to the LLM.

Example scientific questions and estimation how likely they will be answered correctly by the LLM * Which resources do mention the compound ? >Difficult, because the resources are not really associated with the compounds * Count how many entries per resource report about > same as the prior * How many pathways between barley and wheat are shared, how many differ significantly based on involved compounds? Order them based on the number of differences > Should be manageable for an LLM to answer based on the KG * Which authors were involved in multiple studies about barley vrs1 gene analysis? > gene information is mentioned in description part of the edal system, this heavily relies on NLP capabilities of the LLM to find that connection and searching for that * What reaction can produce compound in plants? > That should be fairly easy to answer (taxonRange is the connecting edge between taxon and compound)

Outcomes

Approach 1: Construction of a BioCypher Knowledge Graph

Advancements in this approach have resulted in the incorporation of five out of the initially defined fifteen node types, namely `protein`, `biochemical_entity`, `molecular_entity`, `dataset`, and `taxon`. The linking of protein and taxon nodes with the relationship `has_taxonomic_range` was successfully achieved through the implementation of a single edge type.

However, a significant challenge was encountered when attempting to translate URIRef objects into human-readable concepts that could be exposed to the large language model. For instance, the conversion of `URIRef("http://www.w3.org/1999/02/22-rdf-syntax-ns#type")` into a simpler term, such as `type`, proved to be a challenging task due to the inability to automate the process.

Approach 2: Connecting Neo4j Knowledge Graph with BioChatter

This approach initially showed faster progress by focusing on rapid integration of the Neo4j knowledge graph with BioChatter(-light) and the LLM. However, the approach subsequently encountered difficulties, as the URIRef objects could not be effectively abstracted by the language model. In the absence of human-readable concepts within the knowledge graph, the connections established lacked meaningfulness and utility.

An alternative solution could be the utilisation of a knowledge graph with human-readable concepts. The configuration of the Neo4j database for this purpose can be achieved through the utilisation of tools such as the `n10s` plugin. The following example illustrates the implementation of such a setup:

```
docker run --net milvus --name <NAME> --publish=7474:7474 --publish=7687:7687 \
 -v $HOME/neo4j-cati/data:/data \
 -v $HOME/neo4j-cati/logs:/logs \
 -v $HOME/neo4j-cati/import:/var/lib/neo4j/import \
 -v $HOME/neo4j-cati/plugins:/var/lib/neo4j/plugins \
 -v $HOME/neo4j-cati/conf:/var/lib/neo4j/conf \
 --env NEO4JLABS_PLUGINS='["apoc", "n10s"]' \
 -e NEO4J_apoc_export_file_enabled=true \
 -e NEO4J_apoc_import_file_enabled=true \
 -e NEO4J_apoc_import_file_use__neo4j__config=true \
 -e NEO4J_dbms_security_procedures_unrestricted=apoc.* \
 -e NEO4J_dbms.unamanaged_extension_classes=semantics.endpoint=/rdf \
 --env NEO4J_AUTH=none --user="$(id -u):$(id -g)" neo4j:4.4.9
```

The `--net milvus` parameter places the Neo4j instance on the same network as BioChatter for seamless connectivity.

Once the Neo4j instance is running, RDF data can be imported as follows: 1. Ensure a unique constraint for resources: `cypher     CREATE CONSTRAINT n10s_unique_uri ON (r:Resource) ASSERT r.uri IS UNIQUE;` 2. Initialize the `n10s` graph configuration: `cypher     CALL n10s.graphconfig.init();` 3. Import RDF/XML data: `cypher     CALL n10s.rdf.import.fetch(<URL>, "RDF/XML");` 4. Alternatively, for Turtle files: `cypher     CALL n10s.nsprefixes.add(<NAME>, <URL to Vocabulary>);     CALL n10s.rdf.import.fetch(<URL>, "Turtle");` 5. In the `docker-compose.yml` file for `biochatter-light`, ensure the following environment settings are configured: `yaml     environment:     NEO4J_DBNAME: <your_database_name>     KNOWLEDGE_GRAPH_TAB: true     OPENAI_KEY: <your_openai_key>` ## Future Work In light of the outcomes of this use case, several key areas for future work have been identified:

Prior to integration with BioChatter, Approach 1 requires a comprehensive representation of the knowledge graph within the BioCypher framework. This involves expanding the current implementation to include all node types and edge relationships defined in the GraphDB knowledge graph. Furthermore, ensuring that the graph is robustly modelled and fully functional within BioCypher will provide a robust foundation for subsequent integration with natural language interfaces.

The difficulties encountered in Approach 2 emphasise the necessity to concentrate on the manner in which nodes and edges are defined and linked within the knowledge graph. The inability of the LLM to abstract or understand URIRef objects emphasises the importance of designing a knowledge graph with human-readable concepts. It would be beneficial for future work to dedicate time and effort to refining the structure and semantics of the graph, with the aim of ensuring that it is both machine-readable and conceptually intuitive.

A significant aspect to be considered in the design of nodes and edges is their compatibility with LLMs (i.e. how the LLM might interpret the connections). This includes the development of a standardised approach for mapping URIRefs to simplified, human-readable terms that can be directly consumed by LLMs. The experimentation with tools and workflows for generating such mappings automatically may also prove valuable in reducing the manual effort required.

GitHub repository

- https://github.com/sebeier/chem_plant_kg_biocypher

Tomato Knowledge Graph use case

The tomato, *Solanum lycopersicum*, is a model organism of great agro-economic interest. As a sessile organism, the tomato plant is susceptible to many biotic and abiotic stresses and must defend itself despite its immobility. Many processes are involved in the plant's response to such stresses, including specific molecular interactions and pathways. Studying these interactions globally can be challenging because information is dispersed across multiple databases, often with different identifiers for the same molecules, making searches time-consuming and error-prone. To make the analysis of these processes easier and more robust, we constructed TomTom (manuscript in preparation), a knowledge graph containing multiple molecular interactions in the tomato plant from 10 publicly available databases.

Objectives and scientific questions

Starting with TomTom, an existing Biocypher knowledge graph, the main goal during the workshop was to extend the graph to include Biochatter and to estimate the agent's answer based on the pre-existing schema. This estimate is used to further curate the original schema to make it more understandable to the agent. A more robust agent will make the knowledge graph more accessible to non-specialists and increase its usefulness. Users will be able to ask questions about the graph without having to query it.

Outcomes

We connected Biochatter to TomTom and ran several tests with different types of questions, from simple to more complex, to better understand how the agents worked. However, regardless of the complexity of the question, the model did not perform well and produced several hallucinations. We noticed that the main hallucination was related to the names of the relationships. To fix this, we modified TomTom's original schema to make the relationships more explicit to the agent, i.e. changing "transcription factor regulation" to just "regulates" gave better results. We also tried the Ollama3 model instead of ChatGPT-4o to find the most optimal model to reduce hallucinations. Consistent with the living BioChatter benchmark presented on the website, the ChatGPT-4o model achieved the best performance in our case study.

Future Work

Further curation of the original schema is required to make the relationships more understandable and easier for the agent.

GitHub repository

The TomTom source code will be available soon.

Changes to BioChatter

Building on experiences from the workshop, multiple features will be added to BioCypher and BioChatter. Documentation will be extended and adjusted to different user backgrounds to allow for easier onboarding. Dedicated benchmark questions for the plant science use cases will be added to the BioChatter benchmark to continuously monitor performance.

Knowledge graph schema descriptions

Work on the Breeding API both on the basis of the existing API (via BioChatter API calling module) and a dedicated knowledge graph (KG) revealed an imbalance between the descriptive power of API calling vs. KG querying. The informative descriptions used to define fields of the API in a BioChatter API module are present only in rudimentary form in the KG schema; we transmit information about the source and target types of relationships, but we do not contextualise entities and relationships in the schema configuration. Adding `description` fields to elements of the BioCypher schema configuration would allow more expressive interactions with the LLM via BioChatter, in analogy to the `description` parameter of an API field.

Documentation and templating

User behaviour during the project informed potential changes to the documentation layout (refactoring underway) and project template structure. We will adjust the template to allow faster and easier customisation to the user's knowledge base, and remove pitfalls for potential misunderstanding. This will involve a copier / cookiecutter templating approach, with conversational setup of the project, and factoring out of auxiliary functions for demonstration purposes which are not central to the template's functionality (for example, the random generation of KG content).

A further very general insight was the apparent need to maintain documentation in the places of easiest access. While many functionalities are explained and accounted for in the documentation pages of BioCypher and BioChatter, they were frequently not found by the users. Solving this issue will require a combined approach of establishing a better culture for reading documentation in the scientific community, and pragmatic changes to the code base and examples. In a simple case, this means introducing comments in specific code and configuration files that directly spell out the information from the docs in the place that they are relevant too. While this introduces duplication of documentation and a maintenance burden, it seems the only real short-term solution to the "documentation avoidance" behaviour of many users. The in-code additions of documentation could also link to the actual documentation instead of straight duplicating it.

In addition, it became apparent that creating a dedicated repository that collects adapters from anywhere may be useful for discovery. Although this adds maintenance burdens, we will consider this option critically, as it promises great increases in user-friendliness in the mid- to long-term.

Benchmarks

Adding the BrAPI module to the BioChatter API calling framework is implemented in a new branch and pull request. This branch also introduces a dedicated benchmark for calling the BrAPI endpoint. Other benchmarks for KGs of interest, for instance, the Tomato KG, will be added in a similar fashion.

Discussion

Gathering biologists, bioinformaticians and AI experts has been a very efficient way to greatly increase our knowledge of the capabilities and limitations of BioChatter and BioCypher. We better understood what can be achieved with those technologies, how they can interact with existing knowledge graphs, with APIs or data file sources. We have seen that, for isolated questions, it works well against APIs, but when dealing with complex workflows such as chaining API calls, programmatic approaches are better suited. Indeed, BioChatter cannot easily infer complex multi-step business logic (as previously and currently implemented by highly trained human experts) without overspecifying the chatbot prompt, which is hardly sustainable or stable over time.

Similarly, developing agent-based systems to interact with formal and semantic data definitions is not straight-forward. The main issue is one of alignment, as observed in the Chem and Plant KG use case: since LLMs are trained on a huge amount of human-written text, they struggle with non-human-like text as much as the typical human. Complex identifier systems such as those used in ontologies and the semantic web are not well-represented in LLM training sets and thus do not yield great performance in contextualisation and grounding by the LLM, including the important concept of disambiguation. It seems that, in order to align the LLM system with the user demands, the data source needs to be engineered to cater to the grounding capabilities of the LLM, which is mainly rooted in real-world knowledge (human concepts). BioChatter will be further developed to acknowledge and address this limitation.

Agentic systems are likely to be a large part of future science, but their implementation and validation are still subject to very active research. It will be crucial to remain up-to-date with these developments, ideally reflecting them in our open-source frameworks for the scientific community's benefit.

Future work

There will be further implementation by the different groups in some of their tools and services. Thanks to this workshop, we will have the opportunity to further explore the solutions to answer the scientific questions we have listed.

Acknowledgements

The authors would like to thank ELIXIR for their funding support through the Staff Exchange Scheme. We would like to thanks INRAE Versailles for hosting the workshop and Christina Gacic for the organization support. The authors also express their gratitude towards de.NBI (German Network for Bioinformatics Infrastructure) for funding the travel expenses of the German participants. Additional funding was provided by Fundação para a Ciência e a Tecnologia (Portugal) through the R&D Unit "GREEN-IT - Bioresources for Sustainability" (DOI: 10.54499/UIDB/04551/2020 and DOI: 10.54499/UIDP/04551/2020) and LS4FUTURE Associated Laboratory (DOI: 10.54499/LA/P/0087/2020). We are also grateful to the 2024 NARO-INRAE Joint Linkage Call for the travel funding.

References

Baud Agnès, N. D., Wan Mariène. (2022). Traces of transposable elements in genome dark matter co-opted by flowering gene regulation networks. *Peer Community Journal*, *2*(e14). https://doi.org/10.24072/pcjournal.68 **[cito:citation]**

Bleker, C., Ramšak, Ž., Bittner, A., Podpečan, V., Zagorščak, M., Wurzinger, B., Baebler, Š., Petek, M., Križnik, M., van Dieren, A., Gruber, J., Afjehi-Sadat, L., Weckwerth, W., Županič, A., Teige, M., Vothknecht, U. C., & Gruden, K. (2024). Stress knowledge map: A knowledge graph resource for systems biology analysis of plant stress responses. *Plant Communications*, *5*(6), 100920. https://doi.org/https://doi.org/10.1016/j.xplc.2024.100920 **[cito:citation]**

Konzak, J. Z. T. C. C. (1974). A decimal code for the growth stages of cereals. *Weed Res*, *14*, 415–421. **[cito:citation]**