# Democratizing knowledge representation with BioCypher

Sebastian Lobentanzer, Patrick Aloy, Jan Baumbach, Balazs Bohar, Pornpimol Charoentong, Katharina Danhauser, Tunca Doğan, Johann Dreo, Ian Dunham, Adrià Fernandez-Torras, et al.

# Democratising Knowledge Representation with BioCypher

Sebastian Lobentanzer[1], Patrick Aloy[2], Jan Baumbach[3], Balazs Bohar[4,5], Katharina Danhauser[6,7], Tunca Doğan[8,9], Johann Dreo[10,11], Ian Dunham[12,13], Adrià Fernandez-Torras[2], Benjamin M. Gyori[14], Michael Hartung[3], Charles Tapley Hoyt[14], Christoph Klein[6,7], Tamas Korcsmaros[4,15,16], Andreas Maier[3], Matthias Mann[17,18], David Ochoa[12,13], Elena Pareja Lorente[2], Martin Preusse[19], Niklas Probul[3], Benno Schwikowski[10], Bünyamin Sen[8,9], Maximilian T. Strauss[17], Denes Turei[1], Erva Ulusoy[8,9], Judith Andrea Heidrun Wodke[20], Julio Saez-Rodriguez[1,*]

(Middle authors are listed in alphabetical order.)

[1] Heidelberg University, Faculty of Medicine, and Heidelberg University Hospital, Institute for Computational Biomedicine, Bioquant, Heidelberg, Germany

[2] Joint IRB-BSC-CRG Programme in Computational Biology, Institute for Research in Biomedicine (IRB Barcelona), The Barcelona Institute of Science and Technology, Barcelona, Catalonia, Spain

[3] Institute of Computational Systems Biology, University of Hamburg, Germany

[4] Earlham Institute, Norwich, UK

[5] Biological Research Center, Szeged, Hungary

[6] Institute of Human Genetics, Technical University of Munich, 81675 Munich, Germany

[7] Department of Pediatrics, Dr. von Hauner Children's Hospital, University Hospital, Ludwig Maximilian University of Munich, 80337 Munich, Germany

[8] Biological Data Science Lab, Department of Computer Engineering, Hacettepe University, Ankara, Turkey

[9] Department of Bioinformatics, Graduate School of Health Sciences, Hacettepe University, Ankara, Turkey

[10] Computational Systems Biomedicine Lab, Department of Computational Biology, Institut Pasteur, Université Paris Cité, Paris, France

[11] Bioinformatics and Biostatistics Hub, Institut Pasteur, Université Paris Cité, Paris, France

[12] European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Genome Campus, Hinxton, Cambridgeshire CB10 1SD, UK

[13] Open Targets, Wellcome Genome Campus, Hinxton, Cambridgeshire CB10 1SD, UK

[14] Laboratory of Systems Pharmacology, Harvard Medical School, Boston, USA

[15] Imperial College London, London, UK

[16] Quadram Institute Bioscience, Norwich, UK

[17] Proteomics Program, Novo Nordisk Foundation Center for Protein Research, University of Copenhagen, Copenhagen, Denmark

[18] Department of Proteomics and Signal Transduction, Max Planck Institute of Biochemistry, Martinsried, Germany

[19] German Center for Diabetes Research (DZD), Neuherberg 85764, Germany

[20] Medical Informatics Laboratory, University Medicine Greifswald, Greifswald 17475, Germany

* Corresponding author: pub.saez@uni-heidelberg.de

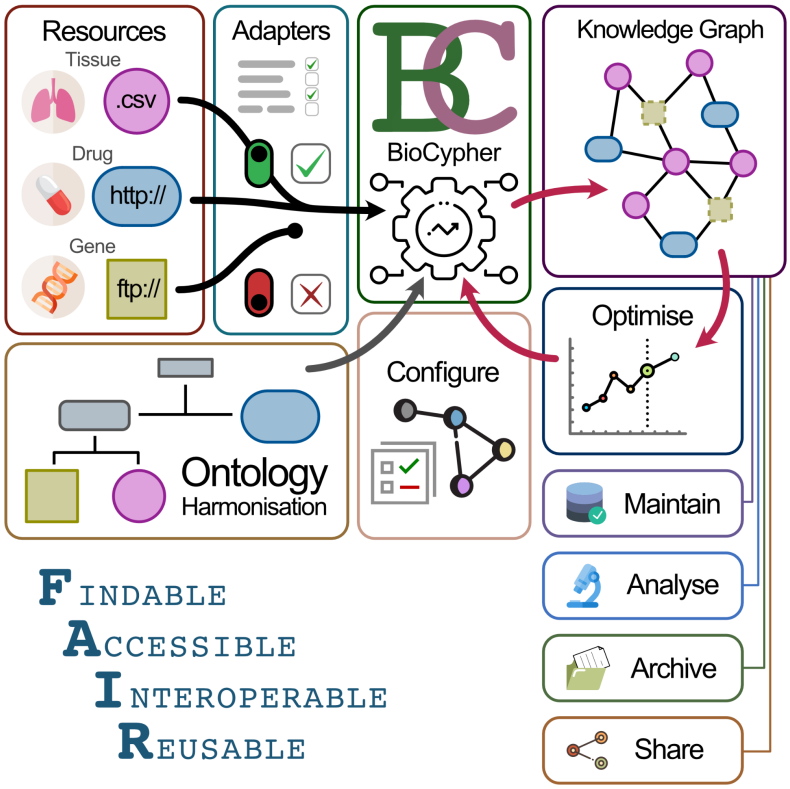
All authors except first and last are listed alphabetically

## Abstract

Standardising the representation of biomedical knowledge among all researchers is an insurmountable task, hindering the effectiveness of many computational methods. To facilitate harmonisation and interoperability despite this fundamental challenge, we propose to standardise the framework of knowledge graph creation instead. We implement this standardisation in BioCypher, a FAIR (findable, accessible, interoperable, reusable) framework to transparently build biomedical knowledge graphs while preserving provenances of the source data. Mapping the knowledge onto biomedical ontologies helps to balance the needs for harmonisation, human and machine readability, and ease of use and accessibility to non-specialist researchers. We demonstrate the usefulness of the framework on a variety of use cases, from maintenance of task-specific knowledge stores, to interoperability between biomedical domains, to on-demand building of task-specific knowledge graphs for federated learning. BioCypher (https://biocypher.org) thus facilitates automating knowledge-based biomedical research, and we encourage the community to further develop and use it.

Keywords: prior knowledge, knowledge graph, database, ontology, harmonisation, federated learning, FAIRness

## Graphical Abstract



## Main Text

## Introduction

Biomedical knowledge, although increasingly abundant, is fragmented across hundreds of resources. For instance, a clinical researcher may use protein information from UniProtKB [1], genetic variants from COSMIC [2], protein interactions from IntAct [3], and information on clinical trials from ClinicalTrials.gov [4]. Combining these complementary datasets is a fundamental requirement for exhaustive biomedical research and thus has motivated a number of integration efforts to form harmonised knowledge graphs (KGs; i.e., knowledge representations based on a machine-readable graph structure). Decisions made on how to store the knowledge at each primary source poses many real-world problems in their recombination, for instance via the use of different identifier namespaces, levels of granularity, or licences [5,6]. However, directly standardising the representation of biomedical knowledge is not appropriate for the diverse research tasks in the community; there is no one-size-fits-all [5–8].

This heterogeneity directly affects the FAIRness of knowledge representation.
**Findability**: Since many KGs have been created, finding the one most suitable for a specific task is challenging and time-consuming [5,6].
**Accessibility**: Few available KG solutions perfectly fit the task the individual researcher wants to perform. Creating custom KGs is only possible for those that can afford years of development time by an individual [7,9] or even entire teams [10]. Smaller or non-bioinformatics labs need to choose from publicly available KGs, thereby also limiting the use of non-public data. There exist frameworks to build certain kinds of KG from scratch [8,11], but these are difficult to use for researchers outside of the ontology subfield and often have a rigid underlying data model [6,12]. Even task-specific knowledge graphs sometimes need to be built locally by the user due to licensing or maintenance reasons, which requires significant technical expertise [13].
**Interoperability**: For the above reasons, many KGs (Supplementary Table 1) are built manually for specific applications, which is very laborious and often redundant, since the primary data sources overlap substantially [5]. For downstream

users, the resulting KGs are too distinct to easily compare or combine [6]. **Reusability**: Maintaining KGs for the community is additional work; once maintenance stops, they quickly deteriorate, leading to reusability and reproducibility issues [5]. Modifying an existing, comprehensive KG for a specific purpose is a non-trivial and often manual process prone to lack of reproducibility[14].

## Approach

To address these problems, we present BioCypher to improve biomedical knowledge representation by means of:

1) **Modularity**: To facilitate the maintenance of multiple task-specific KGs from overlapping primary resources, we propose a modular approach that allows recombining individual data "adapters" for primary resources (e.g. UniProtKB and COSMIC) in a reusable manner. This allows delegating the maintenance work to one central place for each adapter instead of having to maintain the primary resource inside each individual KG (see **case study "Modularity"**).

2) **Harmonisation**: To facilitate harmonisation of datasets from a biological perspective, we propose to use ontology mapping. Primary data sources may represent similar data in different ways. By mapping divergent representations onto the same ontological class (for instance, "protein" or "somatic variant"), harmonisation can be greatly simplified (see **case study "Tumour board"**). In addition, the ontological information projected onto each KG entity allows for more flexible and informative queries in downstream analyses (see **case study "Network expansion"**).

3) **Reproducibility**: By sharing the ontology mapping from (2) in a project-specific manner, a database used for a specific task can be reproduced more effectively. Since sharing the databases themselves is often prohibited by their large size, BioCypher facilitates the creation of task-specific subsets of databases to be shared alongside analyses. This is enabled by extensive automation, reducing the time required and file sizes (see **case studies "Network expansion", "Subgraph extraction", and "Embedding"**).

4) **Reusability and accessibility**: Finally, the sustainability of research software is strongly related to adoption in - and contributions from - the community. We are developing an open source software applying modern methods of continuous integration and development, including many researchers and developers from the beginning (see **case study "Data integration"**). This facilitates robust workflows that are tested end-to-end, including the integrity of the scientific data. We operate under the permissive MIT licence and provide contributors with guidelines for their contributions and a code of conduct. To provide the knowledge and methods to a larger community, we create user-friendly interfaces using open standards. These interfaces, together with the biological perspective introduced by ontologies, improve usability by non-bioinformaticians.

Taken together, these features will make the process of gathering and harmonising biomedical knowledge simpler, more democratic, and FAIR [15].

## Implementation

We build on recent technological and conceptual developments in biomedical ontology that greatly facilitate the harmonisation of biomedical knowledge. We integrate a consistent and comprehensive biomedical ontology, the Biolink model [16], and an extensive catalogue and resolver for biomedical identifier resources, the Bioregistry [17]. Both projects, like BioCypher, are open-source and community-driven. Biolink serves as a biological anchor for the harmonisation of different data sources, and Bioregistry provides consistent vocabularies for representing biological concepts as well as mapping and validation of identifiers. We also facilitate exchange, extension, and modification of the ontological scaffold to accommodate database-specific needs. BioCypher is implemented as a Python library that provides a low-code access point to data processing and ontology manipulation (for examples, see **case studies "Tumour board" and "Network expansion"**). BioCypher facilitates the decision on how to represent knowledge and simplifies the creation of the corresponding KG, bridging the gap between the field of biomedical ontology and the broad application of databases to biological research questions.

BioCypher's translation framework simplifies the creation of custom KGs through a combination of "adapters" (where the data is ingested) and a schema configuration (where the structure of the graph and mappings to ontology are recorded). Building a task-specific KG, given existing configuration, takes only minutes, and creating a KG from scratch can be achieved in a few days of work. This allows for rapid prototyping and automated machine learning (ML) pipelines that iterate the KG structure to optimise predictive performance (for instance, building custom task-specific KGs for graph embeddings and ML (see **case study "Embeddings"**). In spite of its speed, automated end-to-end testing of billions of entities and relationships per KG increases trust in the consistency of the data (see Methods for details and the **case study "Network expansion"** for an example).

By abstracting the KG build process as a combination of modular *input* adapters, BioCypher saves developer time in maintaining integrative resources made up of overlapping primary sources (**Figure 1A**, see **case study "Modularity"**). We are migrating several of these integrative resources using the BioCypher framework, for instance OmniPath [18,19], the Clinical Knowledge Graph (CKG, [20]), CROssBAR v2 [21], the Bioteque [7], and a Dependency Map KG [22]. By mapping each of these knowledge collections onto the same ontological framework, we also gain automatic interoperability between the different biomedical domains (**Figure 1B**).

By providing modular *output* adapters, we can adjust to the various needs of KG users. A Neo4j adapter provides rapid access to extensive databases for querying from analysis (Jupyter) notebooks and facilitates maintenance of large knowledge collections for storage. A CSV writer allows exchange with other knowledge curation services, for instance in the KGX format [12]. Python-native adapters (for instance to sparse matrix or NetworkX format) yield knowledge representations that can immediately be used programmatically in machine learning frameworks such as PyTorch Geometric for deep learning [23].

For high performance, we implement property graph database technologies that provide intuitive query interfaces, such as the Cypher graph query language developed by Neo4j [24]. This enables complex and versatile queries that pave the way towards rich and highly interactive interfaces. For example, web widgets and

apps (such as https://crossbar.kansil.org and https://drugst.one) allow non-computational researchers to browse and customise the database, and to plug it into standard pipelines [25]. Additionally, a structured knowledge representation facilitates connection to modern natural language processing applications [26]. Neo4j is highly scalable and interacts well with other components of large-scale, distributed, high performance computing infrastructure. Thanks to common standards, tools developed can be shared across projects and used community-wide or in cloud-based services that preserve sensitive patient data (see **case study "Federated learning"**).

## Conclusion

Biomedical knowledge is amassed at an ever increasing rate, and machine learning tools that leverage prior knowledge in combination with biomedical big data are gaining much traction, yielding, for instance, sophisticated deep neural architectures that perform prediction of combinatorial perturbations or attempt to diagnose rare diseases [7,27–32]. However, the knowledge representations used in these frameworks result from arbitrary decisions about inclusion and structure, followed by manual implementation, and thus are not optimised for the task at hand nor tested for alternatives or robustness with regard to the representation. BioCypher provides a timely framework for KG standardisation to improve the interoperability of different prior knowledge sources and downstream computational analysis methods. We facilitate FAIRness in knowledge representation by increasing accessibility for non-bioinformatics groups and smaller labs, and demonstrate the key advantages of BioCypher by examples in the Supplementary Note. We invite all database and tool developers to join this collective effort.

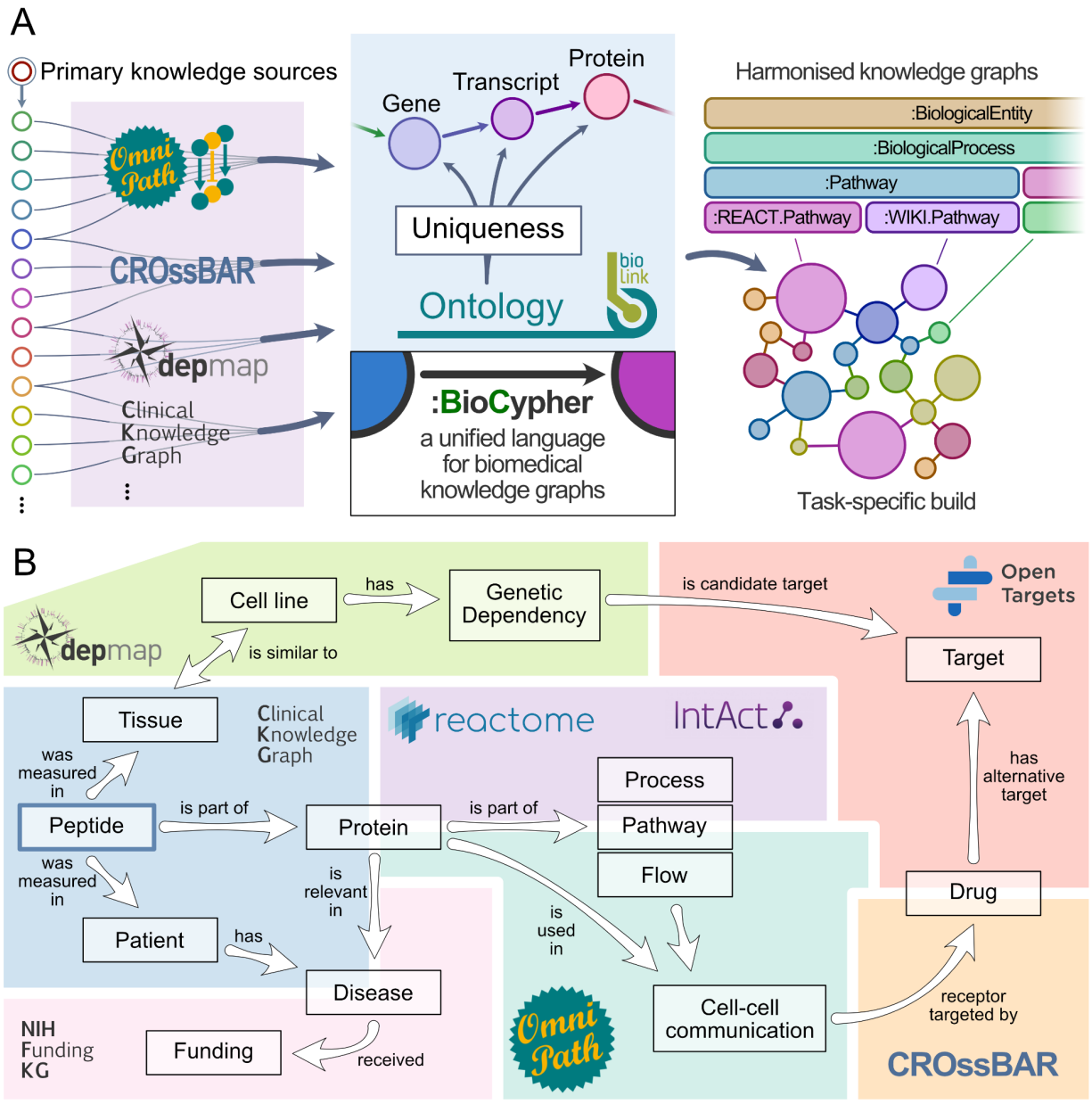



***Figure 1: The BioCypher framework.*** *A) We transform commonly used, curated resources into configurable, task-specific knowledge graphs, using ontology to inform biological "objectness", facilitating integration, reasoning, and interpretation. B) Agreeing on a common representational framework allows recombination of task-specific data sources to answer complex queries across biomedical domains. For instance, starting at mass spectrometry measurements of a patient's tumour (left), one could go through clinical annotations to genetic dependencies from the Dependency Map project to identify potential drug targets, or through pathway / process annotations in Reactome and IntAct, identify relevant ligand-receptor pairs using OmniPath, and use CROssBAR to perform drug discovery or repurposing for these receptors.*

## Methods

BioCypher is implemented as a Python package. Its main purpose is to receive arbitrarily structured biomedical information and create or update a knowledge graph (KG) database that unambiguously maps each KG entity to its corresponding biological class. It uses ontology (the curation of biomedical concepts into a hierarchy of classes) to encode semantic information about KG entities ("objectness") in the biomedical space. From a software development perspective, BioCypher can be described as an extract-transform-load (ETL) pipeline with a focus on biomedicine.

As a baseline, we use the Biolink model [16], a comprehensive and generic biomedical ontology system; where needed, this ontology can be exchanged with or extended by more specific and task-directed ontologies, for instance from the OBO Foundry [33]. Identifier namespaces are collected from the community-curated Bioregistry service [17], where available. Bioregistry also supplies convenient methods for parsing identifier Compact URIs (CURIEs), which are the preferred method of unambiguously specifying identities of KG entities. For identifier mapping, where it is needed, the corresponding facilities of PyPath [18] are used and extended.

The preferred way of entering data into a BioCypher graph attaches scientific provenance to each entry, allowing the aggregation of data with respect to their sources (for instance, the publication an interaction was derived from) and thus avoiding problems such as duplicate counting of the same primary data from different secondary curations. In this way, confidence about knowledge contents of each graph can be assessed more easily, for instance in the order of "multiple curated sources" > "single curated source" > "multiple experimental sources" > "single experimental source" > "predicted interaction". For author attribution, the preferred way of entering data into BioCypher also includes the exact provenance of each entry, for instance, the publication it was derived from or the consortium responsible for the curation of said content. In the same way, all licences of the

contents are propagated forward, enabling the users of the framework to easily determine the allowed uses for any given KG.

Particularly for the creation of databases available to the public we recommend using the "strict mode" of BioCypher, which does not allow creation of entities without associated source, licence, and version parameters. In this scenario, BioCypher can effectively prevent the re-distribution of data whose original licence does not allow it, and guarantees that data originators are acknowledged.

BioCypher is a free software under MIT licence, openly developed and available at https://biocypher.org. Community contributions in the form of GitHub issues or pull requests are very welcome and encouraged.

## Usage

A key advantage of the modular structure of BioCypher is the ability to reuse existing adapters for primary or secondary knowledge sources. In case no adapter exists for a given resource, it can be created following the pattern of one of our existing adapters and shared with the community for further use. To create a custom KG with BioCypher, two main components are necessary: 1) a YAML file (https://yaml.org) detailing the configuration of graph constituents, including their mode of representation (node or edge) and their preferred identifier (default is included); and 2) one or multiple adapter modules responsible for handing off the data to BioCypher. These two components are described below. BioCypher provides a number of utilities for manipulating the input data as well as the ontological foundation of the graph, for instance, filtering properties of input types or arbitrarily extending the ontology. More details and a tutorial can be found in the documentation at https://biocypher.org.

## Schema configuration

Configuration of graph constituents is made available through a graph schema YAML file, whose main purpose is to mediate between the structure of the input data and the resulting BioCypher graph structure. It details, for each constituent species of the graph, the mode of representation (node or edge), the unique

identifier system used (e.g., UniProt or HGNC Symbol), the label to be expected in the input, and - in the case of relationships - the types of source and target nodes. It can also be used to unify the properties attached to nodes and edges in the resulting KG, which is useful when combining sources or dealing with heterogeneous datasets.

### Data retrieval (the adapter)

A Python adapter module is responsible for the actual database creation process. Briefly, the primary data are ingested and passed into BioCypher through the driver instance created in the build pipeline. BioCypher accepts lists and generators; the latter enable streaming of very large datasets that may not fit into the working memory of smaller machines. We provide information about the structure of the input data through the package and its documentation. BioCypher enables automatic extended labelling of each node with the entire hierarchy of that node derived from the ontology tree, which allows more flexibility in querying the resulting KG and simplifies the queries. For instance, a narrow query could yield interactions of proteins, while a query for “polypeptides” (the ontological parent of “protein”) yields proteins, peptides, and precursors; a query for “gene or gene product” additionally returns genes and transcripts without the need for concatenating all individual classes of entities or modifying the underlying graph.

### Interacting with the graph

All interactions with BioCypher take place through its main module (`driver.py`). It connects to a running Neo4j instance with multiple options for authorisation, and also handles calls to the other modules, such as the batch writer used for the rapid `admin import` feature of Neo4j, which allows building and maintaining very extensive databases in very little time. The main modes of graph manipulation are showcased in our tutorial.

The connected graph database can be interactively manipulated through the Python driver supplied by Neo4j through our `neo4j-utils` library (https://github.com/saezlab/neo4j-utils), creating, updating, and deleting nodes and

edges *ad libitum*. BioCypher provides a high-level interface that introduces consistency through the use of standardised structure of the biomedical contents. For internal consistency and integration purposes, each BioCypher graph contains a meta-graph that details its structure and exact settings (e.g., identifiers used), which is created upon instantiation of a graph instance and updated after each manipulation. It includes time-stamps for versioning purposes, which can also be customised with manually created version numbers, facilitating sharing and reproducibility of analyses.

Time can be a limiting factor in the interactive creation of graph content. If the dataset is of large size - say, in the realm of millions of nodes and edges - the creation of a graph can take hours to days. For this reason, we also support the creation of consistent CSV files to be used for the `neo4j-admin import` shell command. This mode significantly speeds up the database creation through the deactivation of safety features that guarantee the consistency of the graph in interactive mode. Thus, we programmatically ensure the consistency of CSV files generated by BioCypher, allowing speedy and secure building of large databases on the fly. Since Neo4j v5, admin import also allows incremental updates of already existing databases, increasing the usefulness of this feature.

## Supplementary Materials

## Case studies

### Modularity

There are several resources used by the biomedical community that can be considered essential to a majority of bioinformatic tasks. A good example is the curation effort on proteins done by the members of the Universal Protein Resource [1]; many secondary resources and tools depend on consistent and comprehensive annotations of the major actors in molecular biology. As such, there is an enormous amount of individual tools and resources that make requests to the public interface of the UniProt service, all of which need to be individually maintained. We and several of our close collaborators make use of this resource, for instance in OmniPath [18], CKG [20], Bioteque [7], and the CROssBAR drug discovery and repurposing database [21]. We have created an example on how to share a UniProt adapter between resources and how to use BioCypher to combine pre-existing databases on the basis of ontology.

We have written such an adapter for UniProt data, using software infrastructure provided by the OmniPath backend PyPath (for downloading and locally caching the data). The adapter provides the data as well as convenient access points and an overview of the available property fields. Using these methods, selecting specific content from the entirety of UniProt data and integrating this content with other resources is greatly facilitated (**Figure S1**), since the alternative would be, in many cases, to use a manual script to access the UniProt API and rely on manual harmonisation with other datasets.

The adapter and a script demonstrating its usage are available at https://github.com/HUBioDataLab/CROssBAR-BioCypher-Migration.

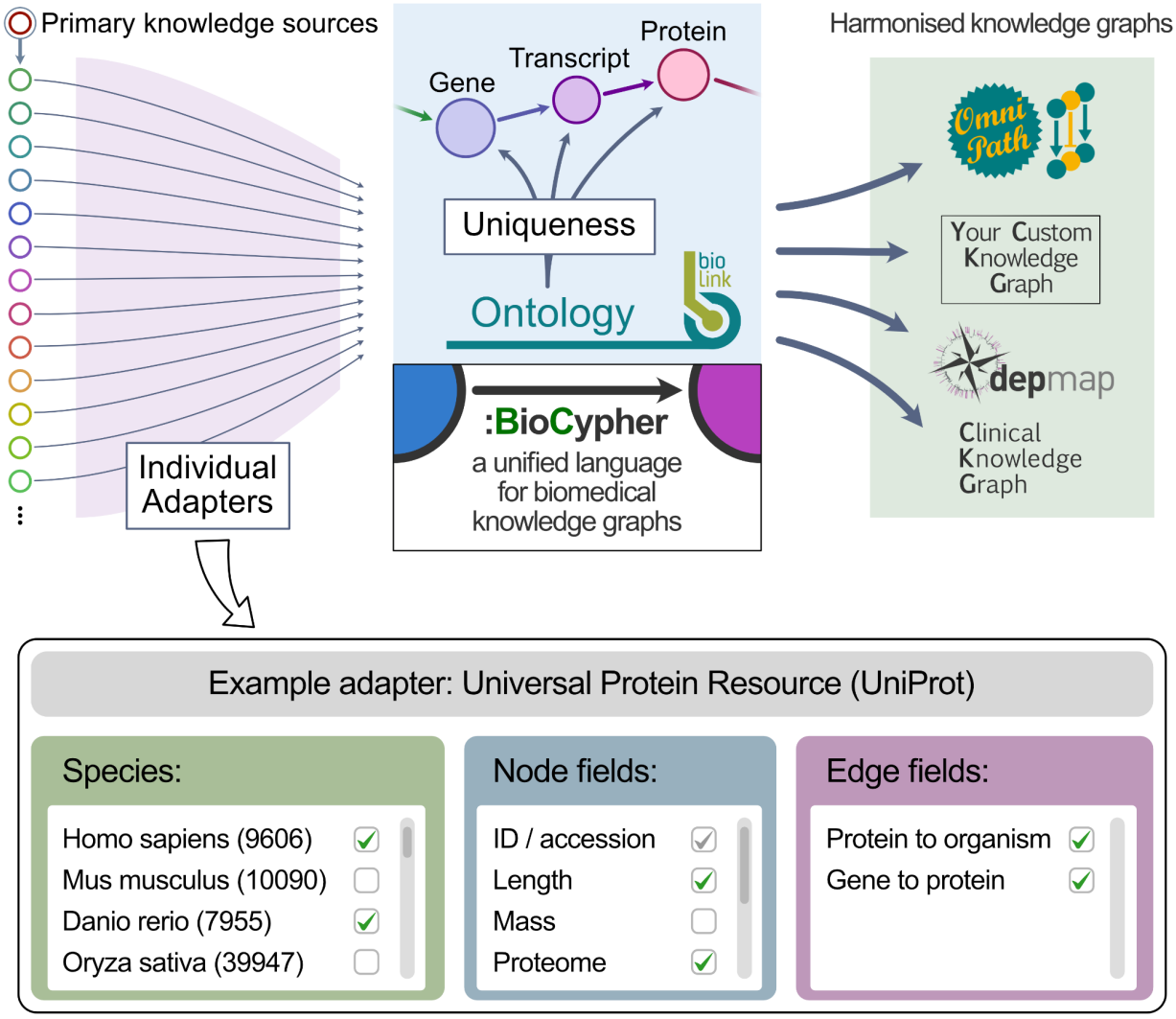



**Figure S1**: **Modularity of knowledge input**. Individual primary source adapters can be used to build secondary knowledge curations such as OmniPath (compare to Figure 1A). This shifts maintenance towards the primary source and thus reduces maintenance effort: instead of maintaining each primary resource at the integrated KG level, only one reusable adapter for each resource is necessary. The primary adapters provide an additional level of flexibility to the user by providing accessible insight into the contents of each primary resource, which can be extensive. For instance, in the adapter for the UniProt knowledge base, the user can select their favourite species, fields of protein information such as the length or mass of the protein, and relationships to import, such as the host organism or the coding gene of each protein.

**Tumour board**

Cancer patients nowadays benefit from a large range of molecular markers that can be used to direct treatment and estimate prognoses [13,34]. In the context of the DECIDER project (www.deciderproject.eu), we are creating a platform to inform the tumour board of actionable molecular phenotypes of high grade serous ovarian cancer patients. The current manual workflow of discovering actionable genetic variants consists of multiple complex database queries to different established cancer genetics databases [13,35,36]. The returns from each of the individual queries then need to be curated by human experts (geneticists) in regard to their identity (e.g. identify duplicate hits from different databases), biological relevance, level of evidence, and actionability. The heterogeneous nature of results received from different primary database providers makes this a time-consuming task.

To facilitate the discovery of actionable variants and reduce the manual labour of human experts, we use BioCypher to transform the individual primary resources into an integrated, task-specific KG. Through mapping of the contents of each primary resource to ontological classes in the build process, we essentially remove the need to manually curate and harmonise the individual database results. This mapping is determined once, at the beginning of the integration process, and results in a BioCypher schema configuration that details the types of entities in the graph (e.g., patients, different types of variants, related treatment options, etc.) and how they are mapped and thus integrated into the underlying ontological framework. As a second step, datasets that are not yet available from pre-existing BioCypher adapters are adapted in similar fashion to yield data ready to be ingested by BioCypher. The code for this project can be found at https://github.com/oncodash/oncodashkb.

We make use of the ontology manipulation facilities provided by BioCypher to extend the broad but basic Biolink ontology at certain branches where it is useful to have more granular information about the data that enters the KG. For example, the exact type of genetic variants are of high importance in the molecular tumour board process, but Biolink only provides a generic “sequence variant” class in its

schema. Therefore, we extended the ontology tree at this node with the very granular corresponding subtree of the Sequence Ontology (SO, [37]), yielding a hybrid ontology with the generality of Biolink and the accuracy of a specialised ontology of sequence variants (**Figure S2**). Due to the mechanism provided by BioCypher, this hybridisation can be performed by providing only the minimal input of the sequence ontology URL and the nodes that should be the point of merging ("sequence variant" in Biolink and "sequence_variant" in SO). The same process is used with the Disease Ontology [38] and OncoTree ([39], see **Figure S2**).

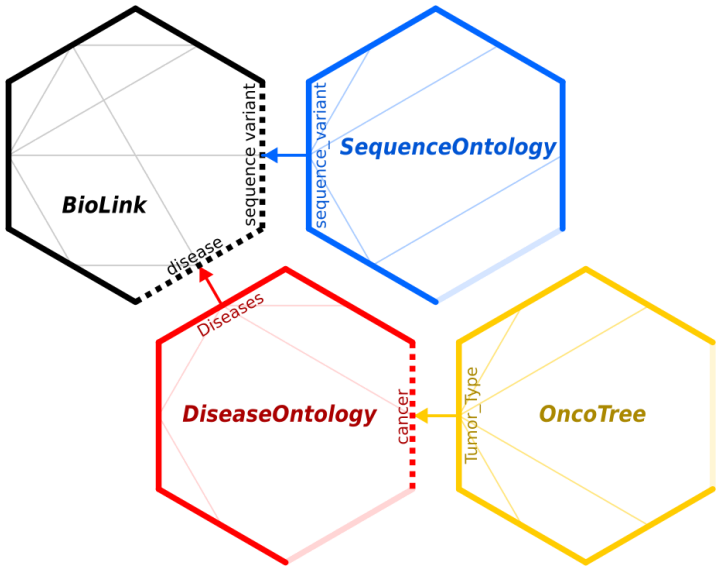



***Figure S2: The ontology manipulation feature is used to extend Biolink with more refined ontologies.*** *Since Biolink has a broad but general representation of biomedical classes, we extend the "sequence variant" with the corresponding granular information from the specialised Sequence Ontology. Similarly, information about cancer and specific tumour types are added from Disease Ontology and OncoTree.*

Once the database has been created through BioCypher, the process of querying for an actionable variant and its associated treatment options for a given patient is greatly simplified. This kind of approach is also known to improve the concordance of knowledge base sources, the ability to incorporate external clinical resources, and the recovering of evidence only represented in a single resource [13].

The major advantage of using BioCypher to integrate several resources is the formal representation of the process provided by the schema configuration, which allows for a simple description and long-term maintenance. Other approaches [13] would need ad-hoc scripts, hindering refactoring if the input resources change, and would lose metadata about the provenance of the merged information, hindering *a posteriori* analysis.

**Network expansion**

Database schemata of large-scale biomedical knowledge providers are tuned for effective storage. For analysis, the user may benefit from a more dedicated schema type corresponding to the biological question under investigation. We created BioCypher with the objective to simplify the transformation from storage-optimised schemas to analysis-focused schemas. Given one or multiple data sources, the user should be able to quickly build a task-specific knowledge graph using only a simple configuration of the desired graph contents. We demonstrate the simplifying capabilities using an interaction-focussed version of the Open Targets graph database as an example [40].

Barrio-Hernandez et al. used this graph database to inform their method of network expansion [41]. The database runs on Neo4j, containing about 9 million nodes and 43 million edges. It focuses on interactions between biomedical agents such as proteins, DNA/RNA, and small molecules. Returning one particular interaction from the graph requires a Cypher query of ~13 lines which returns ~15 nodes with ~25 edges (variable depending on the amount of information on each interaction). A procedure to collect information about these interactions from the graph is provided with the original manuscript [41], containing Cypher query code of almost 400 lines (http://ftp.ebi.ac.uk/pub/databases/intact/various/ot_graphdb/current/apoc_procedures_ot_data.txt). Still, this extensive query only covers 11 of the 37 source labels, 10 of the 43 target labels, and 24 of the 76 relationship labels that are used in the graph database, offering a large margin for optimisation in creating a task-specific KG.

After BioCypher adaptation, the KG (covering all information used by Barrio-Hernandez et al.) has been reduced to ~700k nodes and 2.6 million edges, a more than ten-fold reduction, without loss of information with regard to this specific task. Compared to the original file of the database dump (zipped, 1.1 GB), the BioCypher output is ~20-fold smaller (zipped, 63 MB), which greatly facilitates sharing and accessibility (e.g. by simplifying online access via Jupyter notebooks). The Cypher query for an interaction has been reduced from 13 query lines, 15 nodes, and 25 edges to 2 query lines, 3 nodes, and 2 edges (**Figure S3**).

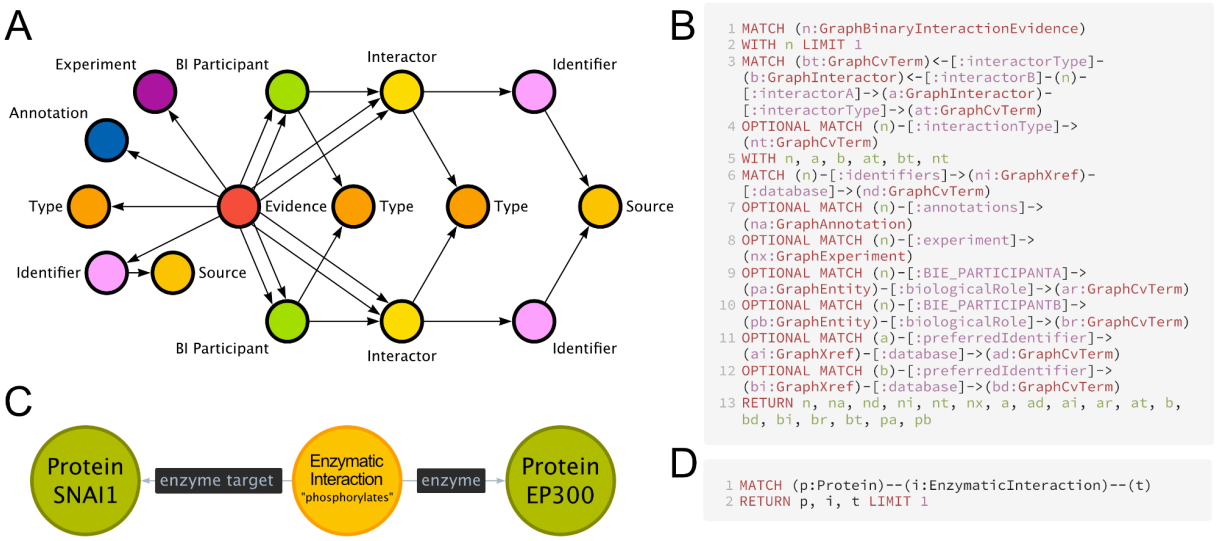



***Figure S3: Semantic abstraction*** *A) The original, "storage-oriented" format used by the OTAR KG, displaying one interaction with additional data. B) The Cypher query to receive one interaction from the OTAR graph. C) The migrated, "task-oriented" format produced by the BioCypher adapter, displaying one interaction. The "additional data" from (A) about experiment and evidence type can be added to the interaction node as a property or encoded in additional nodes connected to the interaction node. D) The Cypher query to receive one interaction from the migrated graph.*

Most of this reduction is due to removal of information that is not relevant to the task at hand and semantic abstraction; for instance, the original chain of `*("hgnc")-[:database]-("SNAI1")-[:preferredIdentifier]-(:Interactor)-[:interactorB]-(:Interaction)-[:interactorA]-(:Interactor)-[:preferredIdentifier]-("EP300")-[:database]-("hgnc")*` to qualify one protein-protein-interaction can be reduced to `*("EP300")-[:enzyme]-("phosphorylation")-[:enzyme target]-("SNAI1")*`. Arguably, the

shorter BioCypher query is also more informative, since it details the type of interaction as well as the roles of the participants. In addition, this representation returns sources of information about the proteins and the interaction as properties on the nodes, and the hierarchical ontology-derived labels provide rich information about the biological context. For instance, the first ancestor labels of the "*phosphorylation*" node are "*enzymatic interaction*", "*direct interaction*", and "*physical association*", grounding this specific interaction in its biological context and enabling flexible queries for broader or more specific terms. This additional information was introduced into the data model by combining the Biolink ontology with the molecular interaction ontology by the Proteomics Standards Initiative [42]. Thus, this "task-oriented" representation is complementary to the "storage-oriented" one, serving a different purpose, and BioCypher provides an easy and reliable way of going from one type of representation to the other.

The BioCypher migration is fast (about 15 minutes on a common laptop) and tested end-to-end, including deduplication of entities and relationships as well as verbose information on violations of the desired structure (e.g., due to inconsistencies in the input data), making the user explicitly aware of any fault points. Through this feedback, several inconsistencies were found in the original Open Targets graph during the migration, some of which originated from misannotation in the Signor primary resource (e.g., "*P0C6X7_PRO_0000037309*" and "*P17861_P17861-2*"). This problem affected only a few proteins, which could have gone unnoticed in a manual curation of the data; a problem that likely is common in our current biomedical knowledge.

Knowledge representations can and should be tuned according to the specific needs of the downstream task to be performed; BioCypher is designed to accommodate arbitrarily simple or complex representations while retaining information important to biomedical research tasks. A compressed structure is important, for instance, in graph machine learning and embedding tasks, where each additional relationship exponentially increases computational effort for message passing and embedding techniques [7,43]. Most importantly, evidence (which experiment and publication the knowledge is derived from) and provenance (who provided which aspects of the primary data) should always be propagated. The

former is essential to enable accurate confidence measures; e.g., not double-counting the same information because it was derived from two secondary sources which refer to the same original publication. The latter is important for attribution of work that the primary maintainers of large collections of biomedical knowledge provide to the community.

The code of this migration can be found at https://github.com/saezlab/OTAR-BioCypher.

**Subgraph extraction**

For many practical tasks in the workflow of a research scientist, the full KG is not required. For this reason, building complex and extensive KGs such as the CKG [20] or the Bioteque [7] would not be sensible in all use cases.

For instance, in the context of a proteomics analysis, the user would only like to contextualise their list of differentially abundant proteins using literature connections in the CKG, rendering much of the information on genetics and clinical parameters unnecessary. In addition, the KG may contain sensitive data on previous projects or patient samples, which cannot be shared (e.g. in the case of publishing the analysis), causing reproducibility issues. Likewise, some datasets cannot be shared due to their licences. With BioCypher, a subset of the entire knowledge collection can be quickly and easily created, taking care to not include sensitive, irrelevant, or unlicenced data. The analyst merely needs to select the relevant species (e.g. proteins, diseases, and articles) and their relationships in the BioCypher configuration. BioCypher then queries the original KG and extracts the required knowledge, conserving all provenance information, and yielding a much reduced data set ready for sharing. Since a complete CKG adapter already existed (found at https://github.com/saezlab/CKG-BioCypher/), the subsetting required minimal effort; i.e., the only required step was to remove unwanted contents from the complete schema configuration. The code for this task can be found at https://github.com/saezlab/CKG-BioCypher/tree/subset.

**Embedding**

As a second subsetting example, we demonstrate the usefulness of subsetting KGs for task-specific graph embeddings. KG embeddings can be an efficient lower-dimensional replacement of the original data in many machine learning tasks[7] and, as methods such as GEARS [27] show, these embeddings can be useful for complex machine learning tasks. However, including all prior data in every embedding is not necessary for good results, while using the proper domain of knowledge can vastly increase the performance of downstream tasks [7]. This issue extends both to the type of knowledge represented (not every kind of relationship is relevant to any given task) as well as the source of the knowledge (different focus points in knowledge resources lead to differential performance across different tasks). Thus, it is highly desirable to have a means to identify the proper knowledge domain relevant to a specific task to increase the efficiency and efficacy of subsequent analysis. To achieve this aim, BioCypher can facilitate task-specific builds of well-defined sets of knowledge from a combination of primary sources for each application scenario. And, since the BioCypher framework automates much of the build process going from only a simple configuration file, the knowledge representations can be iterated over quickly to identify the most pertinent ones. As above, the only requirement from the user (given existing BioCypher adapters for all requested primary sources) is a selection of biological entities and relationships between them in the schema configuration.

**Federated learning**

Federated learning is a machine learning approach that enables multiple parties to collaboratively train a shared model while keeping their data decentralised and private [44,45]. This is achieved by allowing each party to train a local version of the model on their own data, and then sharing the updated model parameters with a central server that aggregates these updates. However, most machine learning algorithms depend on a unified structure of the input; when it comes to algorithms that combine prior knowledge with patient data, a large amount of harmonisation needs to occur before the algorithms can be applied.

BioCypher facilitates federated machine learning by providing an unambiguous blueprint for the process of mapping input data to ontology. Once a schema for a specific machine learning task has been decided on by the organisers, the BioCypher schema configuration can be distributed, ensuring the same database layout in all training instances. For example, the Care-for-Rare project of the Munich Children's Hospital has to synchronise a broad spectrum of biomedical data: demographics, medical history, medical diagnosis, laboratory results from routine diagnostics, imaging and omics data with analyses of proteome, metabolome and transcriptome in different tissues as well as genetic information. To allow reaching a sample size that is suitable for modern methods of diagnosis and treatment options in rare diseases [32], world-wide collaboration between children's hospitals is a necessity. The unstructured nature of most clinical data necessitates a harmonisation step with subtle challenges with respect to ontology. For instance, general classifications such as ICD10-GM subsume rare childrens' diseases under umbrella terms for whole disease groups, requiring alternative coding catalogues such as Orphanet OrphaCodes [46] and the German Alpha-ID [47]. Larger ontologies such as HPO [48] and SNOMED-CT [49] are complex and expanded constantly. In addition to the technical challenges, the legal requirements of patient confidentiality and data protection necessitate extreme care in the processing of all data, hindering information sharing between collaborators. All of the above poses great challenges in data integration in the clinical setting.

Using BioCypher, we enable a federated learning pipeline by supplying build instructions for each local database in the form of the schema configuration. At each location, a task-specific KG is created from public data with the Clinical Knowledge Graph as baseline, using the subsetting facilities described in the case study "Subsetting large datasets". Afterwards, the sensitive patient data (e.g., germ-line genetic variants) are integrated into this KG at each location, using the BioCypher schema configuration to specify the type of data involved (e.g., clinical measurements, genetic profiling). This ensures that, regardless of how the sensitive data are represented at each location, the machine learning algorithm works with the exact same structure of KG, preventing accidental data leakage.

**Data integration**

The German Center for Diabetes Research (DZD, www.dzd-ev.de) has developed a knowledge graph to support data integration for translational research. The internal KG instance provided the foundation of the open-source CovidGraph project [50] which is now maintained by the HealthECCO community (www.healthecco.org). At the core of the DZD KG is a data ingestion pipeline for PubMed that transforms publication data into a detailed graph representation, including authors, affiliations, references, and MeSH term annotations. The PubMed graph contains 350 million nodes and 850 million relationships, as well as data on biological entities (genes, transcripts, proteins), their functional annotations, and biochemical interactions. This KG is used to link internal research data to public knowledge and to generate new research hypotheses.

The growth of the graph posed three major challenges:

1. Maintaining data ingestion pipelines for dozens of upstream data sources is not feasible in a research context.
2. The KG used a custom data model that was able to capture the initial information. The effort to integrate a new upstream data source grows with the total number of data sources. Each new data source has to be cross-referenced with all existing data sources and inconsistencies arise because the same piece of information may be represented with different levels of abstraction.
3. The custom data model complicated the collaboration with external researchers. Integrating data from different contexts required the collaborator to adapt to the internal data model.

BioCypher can handle all three challenges. Firstly, the open architecture and community effort around BioCypher allows maintaining core data ingestion pipelines while reusing data adapters from experts in other fields. Secondly, the well described data model of Biolink drastically reduces the effort required to integrate new data sources because they need only to be adapted to the core data model, not to all existing data. Thirdly, the combination of an open architecture

and ontology-based data integration facilitates collaborations with external researchers.

This approach reduces the time required to bring new data products to researchers at the DZD because the unified data model and ontology-backed data harmonisation allows the reuse of data analysis modules and user interface components. Removing obstacles for collaboration on the knowledge graph supports interdisciplinary research on diabetes complications and comorbidities.

***Supplementary table 1. Knowledge graph solutions (non-comprehensive).***

| Database | Reference |
|---|---|
| Biological Insight Knowledge Graph | 10 |
| Bioteque | 7 |
| Clinical Knowledge Graph | 20 |
| CROssBAR | 21 |
| Dependency Map | 22 |
| GenomicKB | 51 |
| HealthECCO Covidgraph | 50 |
| INDRA CogEx | https://github.com/bgyori/indra_cogex |
| KG-COVID-19 | 12 |
| OmniPath | 18 |
| Open Targets | 40 |
| PheKnowLator | 8 |
| PORI (Platform for Oncogenic Reporting and Interpretation) | 13 |
| PrimeKG | 52 |
| RTX-KG2 | 53 |
| TypeDB | https://github.com/typedb-osi/typedb-bio |

## Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 965193 for DECIDER and No 116030 for TransQST, and the German Federal Ministry of Education and Research (BMBF, Computational Life Sciences grant no. 031L0181B).

We are thankful to Henning Hermjakob, Benjamin Haibe-Kains, Pablo Rodriguez-Mier, Daniel Dimitrov, and Olga Ivanova for feedback on the initial draft of the manuscript.

## Conflict of interests

JSR reports funding from GSK and Sanofi and fees from Travere Therapeutics and Astex Pharmaceuticals.

## Author contributions

SL conceptualised and developed BioCypher and wrote the manuscript. KD and NP contributed to the federated learning case study. JD contributed to the tumour board case study. AFT and EPL contributed to the embedding case study. MP contributed to the data integration case study. BS and EU contributed to the modularity case study. DT contributed to BioCypher development and the manuscript. JSR conceptualised BioCypher, supervised the project, and acquired funding. All coauthors revised the manuscript.

Bibliography


1. UniProt Consortium. UniProt: a hub for protein information. *Nucleic Acids Res.* **43**, D204-12 (2015).
2. Forbes, S. A. *et al.* COSMIC: exploring the world's knowledge of somatic mutations in human cancer. *Nucleic Acids Res.* **43**, D805-11 (2015).
3. Hermjakob, H. *et al.* IntAct: an open source molecular interaction database. *Nucleic Acids Res.* **32**, D452-5 (2004).
4. Zarin, D. A. *et al.* Issues in the registration of clinical trials. *JAMA* **297**, 2112–2120 (2007).
5. Bonner, S. *et al.* A review of biomedical datasets relating to drug discovery: a knowledge graph perspective. *Brief. Bioinformatics* **23**, (2022).
6. Callahan, T. J., Tripodi, I. J., Pielke-Lombardo, H. & Hunter, L. E. Knowledge-Based Biomedical Data Science. *Annu. Rev. Biomed. Data Sci.* **3**, 23–41 (2020).
7. Fernandez-Torras, A., Duran-Frigola, M., Bertoni, M., Locatelli, M. & Aloy, P. Integrating and formatting biomedical data in the Bioteque, a comprehensive repository of pre-calculated knowledge graph embeddings. *bioRxiv* (2022).
8. Callahan, T. J., Tripodi, I. J., Hunter, L. E. & Baumgartner, W. A. A Framework for Automated Construction of Heterogeneous Large-Scale Biomedical Knowledge Graphs. *BioRxiv* (2020) doi:10.1101/2020.04.30.071407.
9. Lobentanzer, S., Hanin, G., Klein, J. & Soreq, H. Integrative transcriptomics reveals sexually dimorphic control of the cholinergic/neurokine interface in schizophrenia and bipolar disorder. *Cell Rep.* **29**, 764-777.e5 (2019).
10. Geleta, D. *et al.* Biological Insights Knowledge Graph: an integrated knowledge

graph to support drug development. *BioRxiv* (2021) doi:10.1101/2021.10.28.466262.

11. Hoyt, C. T. *et al.* Integration of Structured Biological Data Sources using Biological Expression Language. *BioRxiv* (2019) doi:10.1101/631812.
12. Reese, J. *et al.* KG-COVID-19: a framework to produce customized knowledge graphs for COVID-19 response. *BioRxiv* (2020) doi:10.1101/2020.08.17.254839.
13. Reisle, C. *et al.* A platform for oncogenomic reporting and interpretation. *Nat. Commun.* **13**, 756 (2022).
14. Ma, C., Zhou, Z., Liu, H. & Koslicki, D. Predicting Drug Repurposing Candidates and Their Mechanisms from A Biomedical Knowledge Graph. *BioRxiv* (2022) doi:10.1101/2022.11.29.518441.
15. Wilkinson, M. D. *et al.* The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* **3**, 160018 (2016).
16. Unni, D. R. *et al.* Biolink Model: A universal schema for knowledge graphs in clinical, biomedical, and translational science. *Clin. Transl. Sci.* **15**, 1848–1855 (2022).
17. Hoyt, C. T. *et al.* Unifying the identification of biomedical entities with the Bioregistry. *Sci. Data* **9**, 714 (2022).
18. Türei, D., Korcsmáros, T. & Saez-Rodriguez, J. OmniPath: guidelines and gateway for literature-curated signaling pathway resources. *Nat. Methods* **13**, 966–967 (2016).
19. Türei, D. *et al.* Integrated intra- and intercellular signaling knowledge for multicellular omics analysis. *Mol. Syst. Biol.* **17**, e9923 (2021).
20. Santos, A. *et al.* A knowledge graph to interpret clinical proteomics data. *Nat.*

*Biotechnol.* **40**, 692–702 (2022).

21. Doğan, T. *et al.* CROssBAR: comprehensive resource of biomedical relations with knowledge graph representations. *Nucleic Acids Res.* **49**, e96 (2021).
22. Pacini, C. *et al.* Integrated cross-study datasets of genetic dependencies in cancer. *Nat. Commun.* **12**, 1661 (2021).
23. Fey, M. & Lenssen, J. E. Fast Graph Representation Learning with PyTorch Geometric. *arXiv* (2019) doi:10.48550/arxiv.1903.02428.
24. Rodriguez, M. A. & Neubauer, P. The Graph Traversal Pattern. *arXiv* (2010) doi:10.48550/arxiv.1004.1001.
25. Carvunis, A.-R. & Ideker, T. Siri of the cell: what biology could learn from the iPhone. *Cell* **157**, 534–538 (2014).
26. Castelvecchi, D. Are ChatGPT and AlphaCode going to replace programmers? *Nature* (2022) doi:10.1038/d41586-022-04383-z.
27. Roohani, Y., Huang, K. & Leskovec, J. GEARS: Predicting transcriptional outcomes of novel multi-gene perturbations. *BioRxiv* (2022) doi:10.1101/2022.07.12.499735.
28. Lotfollahi, M. *et al.* Biologically informed deep learning to infer gene program activity in single cells. *BioRxiv* (2022) doi:10.1101/2022.02.05.479217.
29. Li, M. M., Huang, K. & Zitnik, M. Graph representation learning in biomedicine and healthcare. *Nat. Biomed. Eng.* (2022) doi:10.1038/s41551-022-00942-x.
30. Dugourd, A. *et al.* Causal integration of multi-omics data with prior knowledge to generate mechanistic hypotheses. *Mol. Syst. Biol.* **17**, e9730 (2021).
31. Yasunaga, M. *et al.* Deep Bidirectional Language-Knowledge Graph Pretraining.

*arXiv* (2022) doi:10.48550/arxiv.2210.09338.

32. Alsentzer, E. *et al.* Deep learning for diagnosing patients with rare genetic diseases. *bioRxiv* (2022).
33. Smith, B. *et al.* The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat. Biotechnol.* **25**, 1251–1255 (2007).
34. Nair, M., Sandhu, S. S. & Sharma, A. K. Cancer molecular markers: A guide to cancer detection and management. *Semin. Cancer Biol.* **52**, 39–55 (2018).
35. Tamborero, D. *et al.* Support systems to guide clinical decision-making in precision oncology: The Cancer Core Europe Molecular Tumor Board Portal. *Nat. Med.* **26**, 992–994 (2020).
36. Tamborero, D. *et al.* Cancer Genome Interpreter annotates the biological and clinical relevance of tumor alterations. *Genome Med.* **10**, 25 (2018).
37. Eilbeck, K. *et al.* The Sequence Ontology: a tool for the unification of genome annotations. *Genome Biol.* **6**, R44 (2005).
38. Schriml, L. M. *et al.* The Human Disease Ontology 2022 update. *Nucleic Acids Res.* **50**, D1255–D1261 (2022).
39. Kundra, R. *et al.* Oncotree: A cancer classification system for precision oncology. *JCO Clin. Cancer Inform.* **5**, 221–230 (2021).
40. Koscielny, G. *et al.* Open Targets: a platform for therapeutic target identification and validation. *Nucleic Acids Res.* **45**, D985–D994 (2017).
41. Barrio-Hernandez, I. *et al.* Network expansion of genetic associations defines a pleiotropy map of human cell biology. *BioRxiv* (2021) doi:10.1101/2021.07.19.452924.
42. Hermjakob, H. *et al.* The HUPO PSI's molecular interaction format--a

community standard for the representation of protein interaction data. *Nat. Biotechnol.* **22**, 177–183 (2004).

43. Cappelletti, L. *et al.* GraPE: fast and scalable Graph Processing and Embedding. *arXiv* (2021) doi:10.48550/arxiv.2110.06196.
44. Nasirigerdeh, R., Torkzadehmahani, R., Baumbach, J. & Blumenthal, D. B. On the privacy of federated pipelines. in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* 1975–1979 (ACM, 2021). doi:10.1145/3404835.3462996.
45. Matschinske, J. *et al.* The FeatureCloud AI Store for Federated Learning in Biomedicine and Beyond. *arXiv* (2021) doi:10.48550/arxiv.2105.05734.
46. Weinreich, S. S., Mangon, R., Sikkens, J. J., Teeuw, M. E. en & Cornel, M. C. [Orphanet: a European database for rare diseases]. *Ned Tijdschr Geneeskd* **152**, 518–519 (2008).
47. Weber, S. & Dávila, M. German approach of coding rare diseases with ICD-10-GM and Orpha numbers in routine settings. *Orphanet J. Rare Dis.* **9**, 827 (2014).
48. Robinson, P. N. & Mundlos, S. The human phenotype ontology. *Clin. Genet.* **77**, 525–534 (2010).
49. Donnelly, K. SNOMED-CT: The advanced terminology and coding system for eHealth. *Stud. Health Technol. Inform.* **121**, 279–290 (2006).
50. Gütebier, L. *et al.* CovidGraph: a graph to fight COVID-19. *Bioinformatics* **38**, 4843–4845 (2022).
51. Feng, F. *et al.* GenomicKB: a knowledge graph for the human genome. *Nucleic Acids Res.* (2022) doi:10.1093/nar/gkac957.

52. Chandak, P., Huang, K. & Zitnik, M. Building a knowledge graph to enable precision medicine. *BioRxiv* (2022) doi:10.1101/2022.05.01.489928.
53. Wood, E. C. *et al.* RTX-KG2: a system for building a semantically standardized knowledge graph for translational biomedicine. *BMC Bioinformatics* **23**, 400 (2022).