

An Open-Source Knowledge Graph Ecosystem for the Life Sciences

Authors

Tiffany J. Callahan^{1,2}, Ignacio J. Tripodi³, Adrienne L. Stefanski¹, Luca Cappelletti⁴, Sanya B. Taneja⁵, Jordan M. Wyrwa⁶, Elena Casiraghi^{4,7}, Nicolas A. Matentzoglou⁸, Justin Reese⁷, Jonathan C. Silverstein⁹, Charles Tapley Hoyt¹⁰, Richard D. Boyce⁹, Scott A. Malec¹¹, Deepak R. Unni¹², Marcin P. Joachimiak⁷, Peter N. Robinson¹³, Christopher J. Mungall⁷, Emanuele Cavalleri⁴, Tommaso Fontana⁴, Giorgio Valentini^{4,14}, Marco Mesiti⁴, Lucas A. Gillenwater^{1,15}, Brook Santangelo^{1,15}, Nicole A. Vasilevsky¹⁶, Robert Hoehndorf¹⁷, Tellen D. Bennett¹⁸, Patrick B. Ryan¹⁹, George Hripcsak², Michael G. Kahn¹⁵, Michael Bada^{1,15}, William A. Baumgartner Jr^{1†}, Lawrence E. Hunter^{1,15†}

Affiliations

1. Computational Bioscience Program, University of Colorado Anschutz Medical Campus, Aurora, CO, 80045, USA
2. Department of Biomedical Informatics, Columbia University Irving Medical Center, New York, NY 10032, USA
3. Computer Science Department, Interdisciplinary Quantitative Biology, University of Colorado Boulder, Boulder, CO, 80301, USA
4. AnacletoLab, Computer Science Department, University of Milan, 20122, Italy
5. Intelligent Systems Program, University of Pittsburgh, Pittsburgh, PA, 15260, USA
6. Department of Physical Medicine and Rehabilitation, School of Medicine, University of Colorado Anschutz Medical Campus, Aurora, CO, 80045, USA
7. Division of Environmental Genomics and Systems Biology, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
8. Semanticly Ltd, London, UK
9. Department of Biomedical Informatics, University of Pittsburgh School of Medicine, Pittsburgh, PA, 15206, USA
10. Laboratory of Systems Pharmacology, Harvard Medical School, Boston, MA, 02115, USA
11. Translational Informatics Division, University of New Mexico School of Medicine, Albuquerque, NM, 87131, USA
12. SIB Swiss Institute of Bioinformatics, Basel, Switzerland
13. The Jackson Laboratory for Genomic Medicine, Farmington CT, 06032, USA
14. ELLIS, European Laboratory for Learning and Intelligent Systems
15. Department of Biomedical Informatics, University of Colorado School of Medicine, Aurora, CO 80045, USA
16. Data Collaboration Center, Critical Path Institute, 1840 E River Rd. Suite 100, Tucson, AZ, 85718, USA
17. Computer, Electrical and Mathematical Sciences & Engineering Division, Computational Bioscience Research Center, King Abdullah University of Science and Technology, Thuwal 23955-6900, Kingdom of Saudi Arabia
18. Departments of Biomedical Informatics and Pediatrics, University of Colorado School of Medicine, Aurora, CO 80045, USA
19. Janssen Research and Development, Raritan, NJ 08869, USA

corresponding author(s): Tiffany J. Callahan (tiffany.callahan@cuanschutz.edu)

shared senior authorship: William A. Baumgartner Jr. and Lawrence E. Hunter

Abstract

Translational research requires data at multiple scales of biological organization. Advancements in sequencing and multi-omics technologies have increased the availability of these data but researchers face significant integration challenges. Knowledge graphs (KGs) are used to model complex phenomena, and methods exist to automatically construct them. However, tackling complex biomedical integration problems requires flexibility in the way knowledge is modeled. Moreover, existing KG construction methods provide robust tooling at the cost of fixed or limited choices among knowledge representation models. PheKnowLator (Phenotype Knowledge Translator) is a semantic ecosystem for automating the FAIR (Findable, Accessible, Interoperable, and Reusable) construction of ontologically grounded KGs with fully customizable knowledge representation. The ecosystem includes KG construction resources (e.g., data preparation APIs), analysis tools (e.g., SPARQL endpoints and abstraction algorithms), and benchmarks (e.g., prebuilt KGs and embeddings). We evaluate the ecosystem by surveying open-source KG construction methods and analyzing its computational performance when constructing 12 large-scale KGs. With flexible knowledge representation, PheKnowLator enables fully customizable KGs without compromising performance or usability.

Background & Summary

The worldwide growth of biomedical data is exponential, with the volume of molecular data alone expected to surpass more than four exabytes by 2025.¹ Translational science requires integrating data and knowledge at multiple scales of biological organization. Rapid advancements in sequencing and multi-omics technologies have made tremendous amounts of diverse data available for secondary use.²⁻⁵ Multimodal data capture different views and, when properly combined, help characterize complex systems.⁶ Unfortunately, these data are highly distributed and heterogeneous, can be difficult to access due to licensing restrictions, lack interoperability, and often have inconsistent underlying models or representations, which limit most researchers from fully utilizing them.^{7,8}

Knowledge graphs (KGs) have frequently been used to systematically model and interrogate the biology underlying complicated systems, organisms, and diseases.⁹ For example, Figure 1 provides a high-level overview of the main biomedical concepts needed to model our currently accepted knowledge of the Central Dogma¹⁰ and expanded to include pathways, variants, pharmaceutical treatments, and diseases. In the life sciences, KGs are usually constructed from a wide range of data sources like Linked Open Data,¹¹ ontologies, the scientific literature, data derived from electronic health records, and multi-omics experiments.^{8,12} In the biomedical context, nodes usually represent different kinds of biological entities like genes, proteins or diseases, and edges (or triples) are used to specify different types of relationships that can exist between a pair of nodes (e.g., “interaction”, “substance that treats”). Multiple definitions of KGs have been proposed in the literature, all sharing the assumption that KGs are more than simple large-scale graphs.¹³⁻¹⁵ Existing definitions are best summarized by Ehrlinger's and Wöb's (2016) definition: "A knowledge graph acquires and integrates information into an ontology and applies a reasoner to derive new knowledge".¹³ We provide an alternative definition and consider a KG a graph-based data structure representing a variety of heterogeneous entities and multiple types of relationships between them and serving as an abstract framework that is able to infer new knowledge (as well as reveal and resolve discrepancies or contradictions) to address a variety of applications and use cases.

KG construction is not a simple process, requiring significant data preprocessing or wrangling before edge lists can be assembled. Fortunately, several methods have been developed to tackle the primary challenges faced when constructing a KG, including: the integration or harmonization of disparate resources (e.g., SPOKE¹⁶, RTX-KG2¹⁷, Petagraph¹⁸, Bio2RDF¹⁹, and Hetionet²⁰), processing and formatting of structured data and KGs (e.g., Dipper²¹, the Knowledge Graph Exchange [KGX]²²), enhancement or extraction of relationships (e.g., Biomedical Knowledge Discovery Engine [BioKDE]²³, KG-COVID-19²⁴) and evidence (e.g., PrimeKG²⁵) from the literature, and the exchange or sharing of constructed KGs (e.g., Network Data Exchange [NDEX]²⁶, KGX²²). Recently, several frameworks like KG-HUB²⁷, the Clinical KG [CKG]²⁸, RTX-KG2¹⁷, BioCypher²⁹, and the Knowledge Base Of Biomedicine (KaBOB)⁷ which provide all of the aforementioned functionalities have been developed. While methods have been developed for each of the processes or steps required to construct KGs, robust tools and resources to evaluate constructed KGs are lacking.⁸ Traditionally, evaluation of constructed KGs has been task- or domain-specific and largely limited to case studies^{16,17,20,24,25,28,29}. Ideally, constructed KGs would be evaluated in the same manner as other network science (e.g., community detection and link prediction algorithms) and KG or node embedding methods using benchmarks like Zachary's Karate Club graph³⁰, DBPedia³¹, and OpenBioLink³². KG benchmarks could be used to assess the computational performance of KG construction methods and to evaluate the implications of

different knowledge representations on specific tasks. To the best of our knowledge, there are no existing benchmarks to systematically evaluate knowledge representation.

Tackling complex problems within the life sciences requires flexible knowledge representations. An important limitation of existing KG construction methods is fixed or limited flexibility in the way that knowledge is modeled. Within the biomedical domain, knowledge is typically modeled in one of three ways (Figure 2), though the nomenclature used to describe these different approaches differs widely in the literature. For simplicity's sake, we will refer to the three different approaches as simple, hybrid, and complex. The first approach results in a simple graph (Figure 2a). Simple graphs are the most common type of network used in the literature. Example simple graphs include Zachary's Karate Club graph³⁰, Hetionet,²⁰ and SPOKE.¹⁶ In these graphs, entities are represented as nodes, and edges are used to model relationships between them. These graphs usually lack formal semantics for the edges and nodes. Edges are often semantically overloaded, for example ignoring the distinction between data (e.g., a protein participating in a process) and metadata (e.g., the source of information about the protein's participation in that process). Simple graphs are usually straightforward to construct and can be stored as key-value pairs resulting in small file sizes and modest memory. Disadvantages of simple graphs include ad hoc semantics, which decreases interoperability, and a lack of clear specification, making machine inference difficult. The second approach results in a hybrid or property graph (Figure 2b). Example hybrid graphs include KG-COVID-19²⁴, DisGeNET³³, OpenBioLink³², Petagraph¹⁸, the Monarch KG³⁴, and Bio2RDF¹⁹. Hybrid graphs aim to model entities and their relations using a mix of standard network representations and formal semantics, usually the Resource Description Framework (RDF)³⁵ and RDF Schema (RDFS)³⁶. Compared to simple graphs, standards-based hybrid graphs facilitate integration with other resources³⁷ and are more amenable to automated inference. They also provide faceted querying as nodes and edges are typed. One cost of hybrid graphs is that they require substantially more space to store than simple graphs. The third approach results in a complex graph, such as KaBOB⁷, often built on the Web Ontology Language (OWL)³⁸ standard (Figure 2c). Complex graphs are more expressive, facilitating the generation of new knowledge via deductive inference.³⁹ By enforcing explicit semantics, OWL provides advantages over RDF/RDFS in the integration of large biomedical data.⁴⁰ Complex graphs are fully machine-readable, highly expressive, and, because they are built on Description Logics,³⁹ are able to leverage reasoners to verify their logical consistency and do deductive inference. Unfortunately, complex graphs are very large, can be difficult for humans to understand and have been shown to perform poorly on some inductive inference tasks.⁴¹ To date, there are no existing KG construction methods that enable the construction of multiple or alternative versions of the same KG utilizing different underlying knowledge representations, making comparisons and benchmarking difficult.

To address the lack of relevant benchmarks and flexibility in knowledge representation, we developed PheKnowLator (Phenotype Knowledge TransLator), a semantic ecosystem for automating the FAIR (Findable, Accessible, Interoperable, and Reusable)⁴² construction of ontologically grounded KGs with fully customizable knowledge representation. The ecosystem consists of three components (Figure 3): (1) **KG Construction Resources**, which consist of tools to download and process heterogeneous data and algorithms to construct custom KGs; (2) **KG Benchmarks**, which consist of prebuilt KGs that can be used to systematically assess the effects of different knowledge representations on downstream analyses, workflows, and learning algorithms; and (3) **KG Tools** to analyze KGs, including Jupyter Notebook-based use cases and tutorials, cloud-based data storage, application programming interfaces (APIs), and triplestores.

We evaluate the PheKnowLator ecosystem by systematically comparing its components with existing open-source KG construction software using a survey developed to assess the functionality, availability, usability, maturity, and reproducibility of KG construction software. We also assessed the computational performance of the ecosystem when used to construct 12 benchmark KGs designed to provide alternative representations for modeling the molecular mechanisms underlying human disease.

Results

PheKnowLator is open-source and available through GitHub (<https://github.com/callahantiff/PheKnowLator>) and PyPI (<https://pypi.org/project/pkt-kg>). Important manuscript definitions are provided in Supplementary Table 1, acronyms are provided in Supplementary Table 2, and PheKnowLator ecosystem resources are listed in Supplementary Tables 3 and 4.

Evaluation

The PheKnowLator ecosystem was evaluated in two ways. First, publicly available software to construct biomedical KGs was identified and systematically compared using a survey developed to assess the functionality, availability, usability, maturity, and reproducibility of each method. Second, the computational performance of the ecosystem was assessed when used to construct 12 benchmark KGs designed to provide alternative representations for modeling the molecular mechanisms underlying human disease. The resources used for each task are listed in Supplementary Table 4.

Systematic Comparison of Open-Source KG Construction Software

Open-source biomedical KG construction methods available on GitHub were identified and compared to the PheKnowLator ecosystem. A survey was used to compare the methods for the task of constructing biomedical KGs and consisted of 44 questions designed to assess five criteria: KG construction functionality, maturity, availability, usability, and reproducibility (Supplementary Table 5). Of the 1,905 repositories identified on GitHub, 231 contained course, tutorial, or presentation material (i.e., manuscript reviews and slide decks), 278 were duplicate or cloned repositories, 79 were KG applications or services, 60 were websites or resource lists, and 1,253 were determined to be irrelevant (i.e., mislabeled, not biomedical, or not a KG construction method). This initial list was supplemented with 11 methods identified through a review article⁸. The final list included the 15 methods (Table 1 with additional details provided in Supplementary Table 6): Bio2Bel,⁴³ Bio2RDF,⁴⁴ Bio4J,⁴⁵ BioGrakn,⁴⁶ the Clinical Knowledge Graph,⁴⁷ COVID-19-Community,⁴⁸ Dipper,²¹ Hetionet,⁴⁹ IASIS Open Data Graph,⁵⁰ KG-COVID-19,⁵¹ KaBOB,⁵² KGX,²² the Knowledge Graph Toolkit,⁵³ ProNet,⁵⁴ and the SEmantic Modeling machine⁵⁵. The methods are visualized by date of GitHub publication in Figure 4a.

The average coverage score of the five assessment criteria was 3.93 (min=2.79, max=4.90). The coverage of each assessment criterion by method is shown in Figure 4b. Examining the results by assessment criteria revealed interesting patterns. **KG Construction Functionality** (Supplementary Table 7): The majority of the methods (81.3%; n=13) included functionality to download data, while 31.3% (n=5) were able to process free-text and 37.5% (n=6) were able to process clinical data. **Availability** (Supplementary Table 8): Three-fourths of the methods (75%; n=12) were written in Python and 18.8% (n=3) were written in a Java-based language. All of the methods but one were licensed with GPL, MIT, or BSD-3. **Usability** (Supplementary Table 9): Sample data were provided by 94.4% (n=17) of the methods, and 80% (n=14) provided tutorials via R Markdown or

Jupyter Notebook. **Maturity** (Supplementary Table 10): On average, the number of commits per year ranged from 17 to 1,000. Over half of the methods (68.8%, n=11) had been published, and 43.8% (n=7) provided collaboration guidelines. **Reproducibility** (Supplementary Table 11): Tools to enable reproducible workflows and aid in installing the method were provided by 75% (n=12) of the methods. Most often, these tools included Docker containers (n=6) and Jupyter or R Notebooks (n=7), and more than 37.5% (n=6) of the methods used a dependency management program like PyPI or CRAN.

While the PheKnowLator ecosystem was comparable to the other methods on the assessed criteria, we found it to have three important differentiating factors relative to the other methods: (i) tools to assess the quality of underlying ontologies; (ii) logging and documentation of metadata including the KG construction process, the data downloaded, the processing steps applied to each data source, and the node and edge types each source contributes to; and (iii) customizable knowledge representation making it possible to take advantage of advanced Semantic Web tools like description logic reasoners (which we have successfully applied in the construction of KGs by the PheKnowLator ecosystem). The ability to generate multiple versions of the same KGs enables the ecosystem to provide benchmark KGs, which can be used to evaluate modeling decisions and to study the impact of knowledge representation on downstream learning. PheKnowLator included all functionality in the five assessment criteria except for tools to process clinical data, which only 37.5% (n=6) of the methods provided.

Human Disease Knowledge Graph Benchmark Comparison and Construction Performance

The PheKnowLator ecosystem enables users to fully customize KG construction by providing the following parameters (described in detail in the *Construct Knowledge Graphs* section of **Component 1: Knowledge Graph Construction Resources** in the Methods): knowledge model (i.e., complex graphs using class- or instance-based knowledge models), relation strategy (i.e., standard directed relations or inverse bidirectional relations), and semantic abstraction (i.e., transformation of complex graphs into hybrid graphs) with or without knowledge model harmonization (i.e., ensuring a hybrid KG is consistent with the class- or instance-based complex graph it was abstracted from). These parameters enable 12 different versions or benchmarks of each KG build. Descriptive statistics and computational performance of the PheKnowLator ecosystem was assessed when used to build a large-scale heterogeneous KG designed to represent the molecular mechanisms underlying human disease and its 12 associated benchmark KGs (referred throughout the remainder of manuscript as the PKT [PheKnowLator] Human Disease benchmark KGs).

Benchmark Comparison

Under the advice of domain experts (ALS, IJT, LH, and CJM), the PKT Human Disease benchmark KGs were constructed from 12 OBO Foundry ontologies, 31 Linked Open Data sets, and results from two large-scale molecular experiments (Supplementary Table 12). The knowledge representation used for the build is shown in Supplementary Figure 1. A simplified overview of this knowledge representation is provided in Figure 5, which highlights the connectivity between the 12 OBO Foundry ontologies (Figure 5a) and their relationship to the primary node types. The 18 primary node types are listed in Table 2 (visualized in Figure 5b), and 33 primary edge types are shown in Table 3. The primary node and edge types do not include all possible node and edge types made available in the core set of 12 OBO Foundry ontologies, only those that are explicitly modeled in our knowledge representation.

Descriptive statistics for the OBO Foundry ontologies, pre- and post-data quality assessment, are

shown in Table 4 (and detailed statistics provided in Supplementary Table 13). Please note when reporting results, we will refer to edges as triples but they both refer to node-relation-node statements. The size of the ontologies varied widely with the Chemical Entities of Biological Interest (ChEBI)⁵⁶ containing the largest number of triples (n=5,190,458) and the Protein Ontology (PRO; modified to exclude all non-human proteins)⁵⁷ containing the most classes (n=148,243). The Relation Ontology (RO)⁵⁸ contained the fewest triples (n=34,901) and the Sequence Ontology (SO)⁵⁹ contained the fewest classes (n=2,569). The merged set of cleaned OBO Foundry ontologies (i.e., core OBO Foundry ontologies; for additional detail on the ontology cleaning process, please see the Component 1: Knowledge Graph Construction Resources section of the **Methods**) contained 545,259 classes and 13,748,009 triples. Statistics for triples added to the core OBO Foundry ontologies are listed by edge type in Table 5. The largest edge sets were protein-protein (n=618,069 triples), transcript-anatomy (n=439,917 triples), and disease-phenotype (n=408,702 triples). The smallest edge sets were biological process-pathway (n=665 triples), gene-gene (n=1,668 triples), and protein-cofactor (n=1,961 triples).

Descriptive statistics for the 12 PKT Human Disease benchmark KGs are shown in Table 6. The PKT Human Disease benchmark KGs constructed using the class-based knowledge model with inverse relations and without semantic abstraction were the largest (13,803,521 nodes; 41,116,791 triples). All of the PKT Human Disease benchmark KGs built without semantic abstraction, regardless of the knowledge model or relation strategy, contained two connected components and three self-loops. All of the PKT Human Disease benchmark KGs were highly sparse with the average density⁶⁰ ranging from 2.16×10^{-7} to 3.50×10^{-7} and 3.03×10^{-7} to 3.40×10^{-7} for benchmark KGs constructed using class-based and instance-based knowledge models, respectively. When applying semantic abstraction, the PKT Human Disease benchmark KGs constructed using instance-based knowledge models (743,829 nodes; 4,967,391 to 9,624,232 triples) were on average larger than those constructed using the class-based knowledge models (743,829 nodes; 4,967,427 to 7,629,599 triples). All PKT Human Disease benchmark KGs constructed using the instance-based knowledge model with semantic abstraction, regardless of the relation strategy employed, were larger, had a higher average degree, and contained more self-loops when knowledge model harmonization was applied. The average density (6.68 standard relations; 10.26 inverse relations) and number of self-loops (445 standard and inverse relations) did not differ for the PKT Human Disease benchmark KGs constructed using the class-based knowledge model with semantic abstraction and when applying knowledge model harmonization. The PKT Human Disease benchmark KGs constructed with semantic abstraction, with and without knowledge model harmonization, are visualized in Figure 6.

Construction Performance

Performance metrics by KG construction step for each of the 12 PKT Human Disease benchmark KGs are shown in Supplementary Figure 2. On average, **Step 1 (Data Download)** took 2.30 minutes (1.80-3.72 minutes) and used an average of 7.93 GB of memory (7.86-7.99 GB). **Step 2 (Edge List Creation)** took an average of 4.82 minutes to complete (4.80-4.87 minutes) and used an average of 39.55 GB of memory (38.93-40.43 GB). **Step 3 (Graph Construction)** took an average of 391.56 minutes (6.53 hours) to complete (265.98-615.92 minutes; 4.43-10.27 hours) and used an average of 118.69 GB of memory (104.30-147.10 GB). On average, the PKT Human Disease benchmark KGs constructed using class-based knowledge models took roughly the same amount of time and used roughly the same maximum amount of memory as those constructed using instance-based knowledge models. Additionally, regardless of the knowledge model, on average, the PKT Human Disease benchmark KGs built using inverse relations and semantic abstraction took longer to run and required more memory.

Discussion

In this paper we have presented PheKnowLator, a semantic ecosystem for automating the FAIR construction of ontologically grounded KGs with customizable knowledge representation. The ecosystem includes KG construction resources, analysis tools (i.e., SPARQL endpoints and cloud-based APIs), and benchmarks (i.e., prebuilt KGs in multiple formats and embeddings). PheKnowLator enables users to build complex KGs that are Semantic Web-compliant and amenable to automatic OWL reasoning, conform to contemporary graph standards, and are importable by popular graph toolkits. By providing flexibility in the way KGs are constructed and generating multiple types of KGs, PheKnowLator also enables the use of cutting-edge graph-based learning and sophisticated inference algorithms. We demonstrated PheKnowLator's utility by comparing its features to 14 existing open-source KG construction methods and through analyzing its computational performance when constructing 12 different large-scale heterogeneous benchmark KGs. Comparing these methods to PheKnowLator revealed similarities, but also highlighted important differentiating factors lacking in other systems, namely: (1) tools to assess the quality of ontologies (which identify, repair, and document syntactic and semantic errors); (2) logging and metadata documentation (which enable users to quickly debug errors and ensures builds can be rigorously reproduced); and (3) customizable knowledge representation and benchmarks (which enables users to empirically evaluate modeling decisions and find the optimal knowledge model or representation for a particular task). These differences highlight PheKnowLator's ability to provide fully customizable KGs without compromising performance or usability.

One of the biggest challenges to developing novel KG construction methods is properly verifying and robustly validating the resulting KGs. Network-science-based algorithms and machine learning methods typically used within the biomedical domain such as link prediction and knowledge graph embedding are able to make use of well-established benchmarks like YAGO,⁶¹ DBPedia,⁶² and Wikidata,⁶³ which are not specific to the biomedical domain. OpenBioLink³² was developed as a benchmark for biomedical KGs, but is almost exclusively used for link prediction tasks. While it might not be possible to create a universal benchmark to verify or validate biomedical KG construction methods or biomedical KGs, development of trusted resources that are not task-specific (e.g., entity prediction or node classification) would benefit the community. The PheKnowLator ecosystem introduces a set of benchmarks to serve this purpose. These benchmarks were specifically designed to enable two types of tasks: (1) the validation of tools and algorithms designed to analyze KGs (e.g., link prediction algorithms and graph representation learning methods); and (2) the validation and comparison of KGs built using different underlying knowledge representations. The ability to empirically evaluate the pros and cons of different knowledge modeling decisions is important when designing knowledge-based systems⁸ and will become more important as more performant graph representation learning methods are developed, especially with respect to explainability.⁶⁴

PheKnowLator Applications and Use Cases

The majority of existing published KGs and KG construction software within the biomedical domain rely on case studies as a form of evaluation.^{16,18,20,24,25,29} While we did not explicitly include case studies as part of our validation, the PheKnowLator ecosystem has fostered substantial collaborations and led to several publications. PheKnowLator benchmark KGs have been used in applications of toxicogenomic mechanistic inference,⁶⁵ to enable the exploration of large-scale biomedical hypergraphs,⁶⁶ and to facilitate deeper sub-phenotyping of pediatric rare

disease patients.⁶⁷ Recently, PheKnowLator was used to create a disease-specific KG that combined ontology-grounded resources with literature-derived computable knowledge from machine reading.⁶⁸ The resulting KG was then used to identify causal features suitable for addressing confounding bias. PheKnowLator has also been used to generate hypotheses for potential pharmacokinetic natural-product/drug interactions, by facilitating the design and implementation of a KG involving biomedical ontologies, natural-product-ontology extensions, and machine reading from literature.⁶⁹ Finally, the PheKnowLator ecosystem was recently selected as the primary infrastructure to facilitate the development of a large-scale KG (denoted RNA-KG) dedicated to the study and development of RNA-based drugs by integrating more than 50 public data sources.^{70,71} PheKnowLator is also the foundation for novel KG approaches in microbiome research: The microbe-relevant KG Microbe-Gene-Metabolite Link (MGMLink) was constructed by augmenting PheKnowLator with information on microbes from the integrated database gutMGene. GutMGene relationships describing observed microbe-metabolite or microbe-gene associations were introduced to the PheKnowLator knowledge base, enabling a search space for mechanistic understanding of microbial influence on disease at the molecular level.⁷²

In addition to the use of the PheKnowLator KG construction software and benchmark KGs, the ecosystem has also contributed to the development of novel tools and resources. Although results are not yet available, PheKnowLator is currently included in the Continuous Evaluation of Relational Learning in Biomedicine (<https://biochallenge.bio2vec.net/>) task. This task aims to provide a means for evaluating prediction models as new knowledge becomes available over time. Results from this task will provide insight into the usefulness of the PheKnowLator builds and will be used to identify areas where the ecosystem can be improved. Additionally, subsets of prebuilt PheKnowLator KGs have been used to help develop and evaluate novel cutting-edge graph embedding AI tools (i.e., GRAPE⁷³), including random-walk-based embedding methods for extremely large-scale heterogeneous graphs using the PheKnowLator KG builds.⁷⁴ In addition to graph representation learning, prebuilt PheKnowLator KGs were used to prototype a novel method for knowledge-driven mechanistic enrichment of ignored genes (i.e., differentially expressed genes which are associated with a disease experimentally but that have no known association to the disease in the literature).⁷⁵ When applied to preeclampsia, this method was able to identify 53 novel clinically relevant and biologically actionable disease associations. The NIH Common Fund Human BioMolecular Atlas Program (HuBMAP)⁷⁶ needed to assemble a KG based on its own preferred graph schema,⁷⁷⁻⁷⁹ with one focus being to maximize leverage of external references among ontologies for translation. The PheKnowLator ecosystem tool OWL-NETS⁴¹ is currently being used to implement ingestion of other operational ontologies (whether in OWL or not) into HuBMAP and NIH Common Fund Cellular Senescence Network (SenNet)⁸⁰. PheKnowLator was also applied to methods in generating pathway diagrams using biomedically relevant KGs.⁸¹ This novel approach was able to recapitulate existing figures regarding neuroinflammation and Down Syndrome from literature with more detailed and semantically consistent molecular interactions using PheKnowLator.⁸²

Limitations and Future Work

The current work has several important limitations. First, it is important to point out that the systematic comparison we performed of open-source KG construction methods on GitHub was subjective, only included three researchers who are actively involved in the development of PheKnowLator, and was originally performed in 2020. While the results were updated in 2021 and re-reviewed in 2023, it is possible that new methods might not have been included. Further,

only a qualitative comparison was carried out that took into account each method's GitHub and associated publications. Ideally, a fair evaluation would be performed where each method would be downloaded and compared when used to build a KG from the same set of data. Unfortunately, this type of analysis requires significant resources and was not within the scope of our analysis. Second, computational performance metrics were only computed over a single build run due to the amount of resources required to build the KGs. While it is not expected that the results for these metrics would significantly change, small deviations related to data provider constraints with respect to accessing build data could result in different outcomes. Third, we mention that the PheKnowLator ecosystem includes two types of benchmarks: KGs and embeddings. Currently, embeddings are only available for one build (v1.0.0) because the size of the generated KGs were quite small. Subsequent builds have resulted in KGs that are so large that generating embeddings has not been feasible. Fortunately, the recent development of performant embedding tools like GRAPE will enable us to provide embeddings for future builds.⁷³ Fourth, while the ecosystem includes robust logging to monitor metadata and builds, it does not formally integrate resources like the Bioregistry⁸³ and BioLink⁸⁴, which are becoming important new KG standards.^{17,8527} Similarly, the PheKnowLator ecosystem relies heavily on OWLTools⁸⁶ but newer and more stable tools like ROBOT⁸⁷ should be leveraged because it allows for the integration of the OWL API and has improved Jena-based functionality. Fifth, as mentioned above, validating very large KGs, like the ones produced by PheKnowLator, is challenging but important. Additional validation of the PheKnowLator ecosystem, including the construction tools and benchmarks is needed, especially with respect to the different KG builds it produces. Finally, while we have worked hard to ensure that the ecosystem tools and infrastructure are user-friendly, additional work is needed to simplify the inputs and make them more machine-readable (e.g., converting input text files into configurable yaml files) and also develop Graphical User Interfaces for supporting the users in all the steps of KG construction.

Methods

The PheKnowLator ecosystem

The PheKnowLator ecosystem was developed to provide a more comprehensive resource to aid in the construction of KGs within the Life Sciences and consists of three components: (1) **KG Construction Resources**; (2) **Benchmark KGs**; and (3) **KG Tools**. Each component is modular; all features and elements can be replaced or extended as technology evolves or to fit a particular use case. The PheKnowLator ecosystem resources are listed by component in Supplementary Table 3.

Component 1: Knowledge Graph Construction Resources

This component is represented by the largest gray box in Figure 3 and consists of two elements: (1) **Process Data**. Resources to process a variety of heterogeneous data; and (2) **Construct KGs**. An algorithm that enables the construction of different types of heterogeneous KGs. The resources that support these elements are detailed in the *ecosystem Component 1: Knowledge Graph Construction Resources* section of Supplementary Table 3.

Process Data

This element consists of two features and was designed to help users download and prepare a wide variety of heterogeneous data sources needed to construct KGs. The two primary features of this component are: (i) Download and (ii) Preparation.

Download

This feature has been configured to download two types of data: (i) ontologies (e.g., HPO,⁸⁸ GO,⁸⁹ and PRO⁵⁷) and databases (i.e., a data source not represented as an ontology), which includes Linked Open Data (e.g., Comparative Toxicogenomics Database,⁹⁰ UniProt Knowledgebase,⁹¹ STRING⁹²), data from molecular experiments (e.g., the Human Protein Atlas,⁹³ the Genotype-Tissue Expression Project⁹⁴), and existing networks and KGs (e.g., Hetionet,²⁰ the Monarch KG⁹⁵). Ontologies are downloaded using OWLTools⁸⁶ (April 06, 2020 release) and databases are downloaded using a custom-built API capable of processing a variety of file formats (e.g., zip, gzip, tar) from different types of servers and APIs.

Preparation

A collection of tools were developed to help users perform a variety of tasks when preparing data that will be used to construct a KG. This feature provides services to map different types of identifiers (e.g., aligning gene identifiers from the Human Gene Nomenclature Committee [HGNC] to⁹⁶ Entrez Gene⁹⁷ and Ensembl⁹⁸), annotate concepts (e.g., convert strings of tissue names from the Human Protein Atlas⁹³ to Uber-Anatomy Ontology [Uberon]⁹⁹ concepts), filter data (e.g., identify variant-disease relationships from Clinvar¹⁰⁰ with a specific type of experimental validation), and process entity metadata (e.g., obtain PubMed identifiers for exposure-outcome relationships from the Comparative Toxicogenomics Database⁹⁰ and extract synonyms and definitions for OBO Foundry ontology concepts). The Data Preparation Notebook (Data_Preparation.ipynb¹⁰¹) illustrates some of these features. There are also features to assess and repair the quality of OBO Foundry ontologies, which are known to be subject to a variety of errors.^{102–104} The Ontology Cleaning Notebook (Ontology_Cleaning.ipynb¹⁰⁵) includes detailed descriptions and examples of the data quality checks.¹⁰⁶ A report is generated after assessing the quality of each ontology, which provides statistics before and after applying each check (a recent report is available on Zenodo.¹⁰⁷)

Construct Knowledge Graphs

This element consists of four features designed to facilitate the construction of large-scale heterogeneous KGs. Together, these features comprise the core functionality of the PheKnowLator KG construction algorithm (referred to as PKT-KG throughout the remainder of the manuscript). The PKT-KG algorithm requires three input documents: (i) a list of one or more OBO Foundry ontologies; (ii) a list of one or more databases; and (iii) edge list assembly instructions (i.e., instructions for filtering input data sources and references to resources needed to normalize concept identifiers). Additional information on each input is available on GitHub (<https://github.com/callahantiff/PheKnowLator/wiki/Dependencies>). The four primary features of this component are: (i) Edge List Construction, (ii) Ontology Alignment, (iii) Customize Knowledge Representation, and (iv) Output Generation.

Edge List Construction

Using information in the edge list assembly instructions, the edge list construction procedure merges data, applies filtering and evidence criteria, and removes unneeded attributes. To automate this process, we have developed a universal file parser (and constantly update it with procedures for parsing new file types) that currently processes over 30 distinct file types. Once the edge lists are constructed, they are serialized in a JSON file.

Ontology Alignment

OBO Foundry ontologies were selected because they represent canonical knowledge and exist for nearly all scales of biological organization.¹⁰⁸ PKT-KG assumes that every KG is logically grounded¹⁰⁹ in one or more OBO Foundry ontologies. This feature leverages OWLTools⁸⁶ (April 06, 2020 release) to merge the ontologies into a single integrated core ontology.

Customize Knowledge Representation

To enable customization in the way that knowledge is represented when constructing a KG, three configurable parameters are provided:

1. **Knowledge Model.** Following Semantic Web standards,¹¹⁰ PKT-KG defines a KG as $K = \langle T, A \rangle$, where T is the TBox and A is the ABox. The TBox represents the taxonomy of a particular domain.^{111,112} It describes classes, properties/relationships, and assertions that are assumed to generally hold within a domain (e.g., a gene is a heritable unit of DNA located in the nucleus of cells [Figure 7a]). The ABox describes attributes and roles of instances of classes (i.e., individuals) and assertions about their membership in classes within the TBox (e.g., A2M is a type of gene that may cause Alzheimer’s Disease [Figure 7b]).^{111,112} PKT KGs are logically grounded in one or more OBO Foundry ontology.¹⁰⁹ Database entities (i.e., entities from a data source that is not an OBO Foundry ontology) are added to the core OBO Foundry ontologies using either a TBox (i.e., class-based) or ABox (i.e., instance-based) knowledge model. For the class-based approach, each database entity is made a subclass of an existing core OBO Foundry ontology class (see the “Class-based” section of Supplementary Table 14). For the instance-based approach, each database entity is made an instance of an existing core OBO Foundry ontology class (see the “Instance-based” section of Supplementary Table 14). Both approaches require the alignment of database entities to an existing core OBO Foundry ontology class, which is managed by a dictionary that is constructed using tools in the Process Data Element of the **Knowledge Graph Construction Resources** component (subclass_construction_map.pkl¹⁰⁷ [PKT Human Disease KG v2.1.0 May 2021]).
2. **Relation Strategy.** PKT-KG provides two relation strategies. The first strategy is standard or directed relations, through a single directed edge (e.g., “gene causes phenotype”). The second strategy is inverse or bidirectional relations, through inference if the relation is from an ontology like the RO (e.g., “chemical participates in pathway” and “pathway has participant chemical”) or through inferring implicitly symmetric relations for edge types that represent biological interactions (e.g., gene-gene interactions).
3. **Semantic Abstraction.** KGs built using expressive languages like OWL are structurally complex and composed of triples or edges that are logically necessary but not biologically meaningful (e.g., anonymous subclasses used to express TBox assertions with all-some quantification). PKT-KG currently uses the OWL-NETS⁴¹ semantic abstraction algorithm to convert or transform complex KGs into hybrid KGs. OWL-NETS v2.0¹¹³ includes additional functionality that harmonizes a semantically abstracted KG to be consistent with a class- or instance-based knowledge model. For class-based knowledge models, all triples containing *rdf:type* are updated to *rdfs:subClassOf* and for instance-based knowledge models, all triples containing *rdfs:subClassOf* are updated to *rdf:type*. For additional details, see OWL-NETS v2.0 documentation.¹¹³

Output Generation

To ensure features of the Process Data element (**KG Construction Resources** component) are transparent and reproducible, metadata are output for all downloaded (downloaded_build_metadata.txt¹¹⁴) and processed (preprocessed_build_metadata.txt¹⁰⁷) data, including the details of the processing steps applied to each database (edge_source_list.txt¹⁰⁷) and OBO Foundry ontology (ontology_source_list.txt¹⁰⁷). The PKT KG construction process is logged extensively (data processing¹¹⁵ and KG construction¹¹⁵). PKT KGs, including node and relation metadata, are output to a variety of standard formats. A description of all output file types is available from Zenodo.¹¹⁶ Please note that all referenced metadata files are from PKT Human Disease KG builds (v2.1.0 May 2021).

Component 2: Knowledge Graph Benchmarks

This component consists of prebuilt KGs that can be used to systematically assess the effects of different knowledge representations on downstream analyses, workflows, and learning algorithms (Figure 3). Current benchmarks and the features that support them are detailed in the *ecosystem Component 2: Knowledge Graph Benchmarks* section of Supplementary Table 3. Currently, the PheKnowLator ecosystem supports two types of benchmarks: (i) KGs and (ii) embeddings.

Knowledge Graphs

The PKT Human Disease KG was built to model mechanisms of human disease, which includes the Central Dogma and represents multiple biological scales of organization including molecular, cellular, tissue, and organ. The knowledge representation was designed in collaboration with a PhD-level molecular biologist (Supplementary Figure 1). The PKT Human Disease KG was constructed using 12 OBO Foundry ontologies, 31 Linked Open Data sets, and results from two large-scale experiments (Supplementary Table 12). The 12 OBO Foundry ontologies were selected to represent chemicals and vaccines (i.e., ChEBI⁵⁶ and Vaccine Ontology [VO]^{117,118}), cells and cell lines (i.e., Cell Ontology [CL]¹¹⁹, Cell Line Ontology [CLO]¹²⁰), gene/gene product attributes (i.e., Gene Ontology [GO]^{89,121}), phenotypes and diseases (i.e., Human Phenotype Ontology [HPO]⁸⁸, Mondo Disease Ontology [Mondo]¹²²), proteins, including complexes and isoforms (i.e., PRO⁵⁷), pathways (i.e., Pathway Ontology [PW]¹²³), types and attributes of biological sequences (i.e., SO⁵⁹), and anatomical entities (Uberon⁹⁹). The RO⁵⁸ is used to provide relationships between the core OBO Foundry ontologies and database entities. As shown in Figure 5, the PKT Human Disease KG contained 18 node types (Table 2) and 33 edge types (listed by relation in Table 3). Note that the number of nodes and edge types reflects those that are explicitly added to the core set of OBO Foundry ontologies and does not take into account the node and edge types provided by the ontologies. These nodes and edge types were used to construct 12 different PKT Human Disease benchmark KGs by altering the Knowledge Model (i.e., class- vs. instance-based), Relation Strategy (i.e., standard vs. inverse relations), and Semantic Abstraction (i.e., OWL-NETS (yes/no) with and without Knowledge Model harmonization [OWL-NETS Only vs. OWL-NETS + Harmonization]) parameters. Benchmarks within the PheKnowLator ecosystem are different versions of a KG that can be built under alternative knowledge models, relation strategies, and with or without semantic abstraction. They provide users with the ability to evaluate different modeling decisions (based on the prior mentioned parameters) and to examine the impact of these decisions on different downstream tasks.

Embeddings

A modified version of DeepWalk¹²⁴ was used to create node embeddings for the v1.0.0 PKT Human Disease benchmark KGs. Embeddings were trained using 128, 256, and 512 dimensions (i.e., the length of the embedding), 100 walks (i.e., the number of paths generated for each node), a walk length of 20 (i.e., the length or number of nodes included in each path), and a sliding window length of 10 (i.e., the number of nodes to the right and left of the target node, which are used as training data for the target node embedding).

The PKT Human Disease benchmark KGs are built monthly through GitHub Actions-scheduled Cron jobs and implemented using dedicated Docker containers, which output data directly to the PheKnowLator GCS Bucket (<https://console.cloud.google.com/storage/browser/pheknowlator>). The PKT Human Disease benchmark KGs are archived through a dedicated Zenodo Community (<https://zenodo.org/communities/pheknowlator-benchmark-human-disease-kg>).

Component 3: Knowledge Graph Tools

This component consists of tools to analyze and use KGs (Figure 3), which includes Jupyter Notebook-based use cases and tutorials, cloud-based data storage, APIs, and triplestores. The features that support these elements are detailed in the *ecosystem Component 3: Knowledge Graph Tools* section of Supplementary Table 3. The Jupyter Notebooks are available on GitHub and currently include tutorials on using OWL-NETS (OWLNETS_Example_Application.ipynb¹²⁵), querying an RDF KG (RDF_Graph_Processing_Example.ipynb¹²⁶), and searching for paths between two entities in a PKT Human Disease KG (Entity_Search.ipynb¹²⁷). As before, data are publicly available through the PheKnowLator GCS bucket and the PKT Human Disease benchmark KGs Zenodo community. The PheKnowLator ecosystem includes a SPARQL Endpoint (<http://sparql.pheknowlator.com/>). The Database Center for Life Science SPARQL proxy web application¹²⁸ is used as the front end and the data is served from a Blazegraph triplestore.¹²⁹

Fair Data Principles

The PheKnowLator is built on the FAIR principles⁴² (Supplementary Figure 3).⁴² **Findability.** Unique persistent identifiers are used for all data (i.e., downloaded, processed, and generated), metadata (i.e., for all downloaded and processed resources, data quality reports, and logged processes), and infrastructure (i.e., Docker containers, compute instances, and KG builds run via GitHub Actions¹³⁰ and the Google AI Platform¹³¹). All benchmark KGs are built using standardized and persistent node and relation identifiers. **Accessibility.** All data (i.e., downloaded, processed, and generated), constructed KGs, and metadata generated during the KG construction process, are publicly available and accessible via RESTful API access to a dedicated GCS Bucket. Additionally, all builds are versioned on GitHub, Google's Container Registry,¹³² and DockerHub.¹³³ Finally, PheKnowLator provides Jupyter Notebooks and automated dependency generation scripts to improve the usability of its resources. **Interoperability.** The PheKnowLator is built on Semantic Web standards, the KGs benchmarks and construction processes are grounded in OBO Foundry ontologies, and, whenever possible, standard identifiers are assigned for all database resources. Additionally, all constructed KGs and KG metadata are output to a variety of standardized file formats like RDF/XML, N-Triples, JSON, and text files. **Reusability.** Benchmark KGs builds are automated, containerized, and deployed through GitHub Actions workflows, which makes the build process and resulting KGs consistent across versions. Semantic Versioning¹³⁴ is used for all code and documentation. The ecosystem is licensed (Apache-2.0¹³⁵) and all ingested data sources are described transparently on the ecosystem's GitHub Wiki by build version (<https://github.com/callahantiff/PheKnowLator/wiki/Benchmarks-and-Builds>).

Evaluation

The PheKnowLator ecosystem was evaluated in two ways: (1) **Systematic Comparison of Open-Source KG Construction Software**. Publicly available software to construct biomedical KGs was identified and systematically compared using a survey developed to assess the functionality, availability, usability, maturity, and reproducibility of each method. (2) **Human Disease KG Benchmark Comparison and Construction Performance**. The computational performance of the ecosystem was assessed when used to construct 12 benchmark KGs designed to represent the molecular mechanisms underlying human disease. The resources used for each task are listed in Supplementary Table 4.

Systematic Comparison of Open-Source KG Construction Software

A systematic comparison was performed to examine how the PheKnowLator ecosystem compared to existing open-source biomedical KG construction methods available on GitHub. To provide an unbiased comparison, no assumptions were made regarding a specific set of user requirements. Instead, the goal of the comparison was to provide a detailed overview of existing methods. A survey was constructed from five criteria (adapted from the evaluation methodology of Babar et al.¹³⁶) including: KG construction functionality, maturity, availability, usability, and reproducibility. Example questions used to assess each criterion are provided in Supplementary Table 5. The full set of survey questions (n=44) are available as a Google Form.¹³⁷ Existing open-source biomedical KG construction methods were identified by performing a keyword search against the GitHub API. The following words were combined to form 31 distinct keyword phrases, which were queried against existing GitHub repository descriptions and README content: “biological”, “bio”, “medical”, “biomedical”, “life science”, “semantic”, “knowledge graph”, “kg”, “graph”, “network”, “build”, “construction”, “construct”, “create”, “creation”. The GitHub scraper is publicly available as a GitHub Gist and was run in May 2020.¹³⁸ The systematic comparison was completed in May 2020 (and updated in June 2021) by TJC with consultation and oversight from WAB and LEH. The survey was scored out of a total score of five points, which was derived as the sum of the ratio of coverage out of one point for each category: KG Construction Functionality (10 questions); Availability (two questions); Usability (nine questions); Maturity (five questions); and Reproducibility (six questions).

Human Disease Knowledge Graph Benchmark Comparison and Construction Performance

Performance metrics were evaluated when building the PKT Human Disease benchmark KGs (v2.1.0 April 11, 2021; testing version not officially released), which included total runtime (minutes) and minimum, maximum, and average memory use (GB). The PKT Human Disease benchmark KGs (v2.1.0 May 1, 2021) were used to compare builds and produce descriptive statistics. Statistics were calculated to help characterize each benchmark KG, which included counts of nodes, edges or triples, self-loops, average degree, the number of connected components, and the density. The semantically abstracted (with and without knowledge model harmonization) PKT Human Disease benchmark KGs were visualized and examined for patterns. The PKT Human Disease benchmark KGs are available through the PheKnowLator GCS and the PKT Human Disease KG Zenodo Community (v2.1.0_01MAY2021).¹¹⁶

Technical Specifications

The PheKnowLator ecosystem resources, including data used to construct KGs and constructed PKT Human Disease benchmark KGs, and code are listed by component in Supplementary Table 3. The PKT Human Disease KG builds were visualized using Gephi¹³⁹ (v0.9.2). The OpenOrd

Force-Directed layout¹⁴⁰ was applied with an edge cut of 0.5, a fixed time of 0.2, and trained for 750 iterations. To help with interpretation, nodes were colored according to node type. When assessing computational performance, all PKT Human Disease KGs were constructed using Docker (v19.03.8) on a Google Cloud Platform N1 Container-Optimized OS instance configured with 24 CPUs, 500 GB of memory, and a 500 GB solid-state drive Boot Disk. PKT Human Disease KG statistics were calculated using Networkx (v2.4).

Data Availability

The PKT Human Disease KG data, metadata, logs, and KG files are publicly available through the PheKnowLator GCS bucket (<https://console.cloud.google.com/storage/browser/pheknowlator>) and are archived in the PKT Human Disease benchmark KGs Zenodo Community (<https://zenodo.org/communities/pheknowlator-benchmark-human-disease-kg>). A detailed list of all PKT Human Disease benchmark KGs is available on Zenodo (https://zenodo.org/record/7030040/files/full_pheknowlator_build_files.json). Descriptions of the data sources used to build the PKT Human Disease KG are available on the GitHub Wiki: <https://github.com/callahantiff/PheKnowLator/wiki/v2-Data-Sources#data-sources>. The v2.1.0 PKT Human Disease benchmark KGs are available in the PheKnowLator GCS bucket (https://console.cloud.google.com/storage/browser/pheknowlator/archived_builds/release_v2.1.0/build_01MAY2021). The survey used to compare the open-source KG construction resources available on GitHub is accessible from Zenodo (<http://dx.doi.org/10.5281/ZENODO.5790040>).

Code Availability

The PheKnowLator ecosystem coding resources are detailed in Supplementary Table 3 by ecosystem component. The PKT-KG algorithm is publicly available through GitHub (<https://github.com/callahantiff/PheKnowLator>) and PyPI (<https://pypi.org/project/pkt-kg>). The SPARQL Endpoint deployment code and documentation are also available through GitHub: <https://github.com/callahantiff/PheKnowLator/tree/master/builds/deploy/triple-store#readme>. A list of the computational resources used to evaluate the PheKnowLator ecosystem is provided in Supplementary Table 4. The code used to scrape the GitHub API is available as Gist (<https://gist.github.com/callahantiff/0ae1c00df9bec7228be3f6bda5466d73>). The survey of open-source KG construction tools is available on Zenodo (<http://dx.doi.org/10.5281/ZENODO.5790040>). The v2.1.0 PheKnowLator code is available on GitHub (<https://github.com/callahantiff/PheKnowLator/releases/tag/v2.1.0>) and from Zenodo (<https://zenodo.org/record/4685943>).

References

1. Agrawal, R. & Prabakaran, S. Big data in digital healthcare: lessons learnt and recommendations for general practice. *Heredity* **124**, 525–534 (2020).
2. van Dijk, E. L., Auger, H., Jaszczyszyn, Y. & Thermes, C. Ten years of next-generation sequencing technology. *Trends Genet.* **30**, 418–426 (2014).
3. Gupta, N. & Verma, V. K. Next-Generation Sequencing and Its Application: Empowering in Public Health Beyond Reality. in *Microbial Technology for the Welfare of Society* (ed. Arora, P. K.) 313–341 (Springer Singapore, 2019).
4. Graw, S. *et al.* Multi-omics data integration considerations and study design for biological systems and disease. *Mol Omics* **17**, 170–185 (2021).
5. Reuter, J. A., Spacek, D. V. & Snyder, M. P. High-throughput sequencing technologies. *Mol. Cell* **58**, 586–597 (2015).
6. Fröhlich, H. *et al.* From hype to reality: data science enabling personalized medicine. *BMC Med.* **16**, 150 (2018).
7. Livingston, K. M., Bada, M., Baumgartner, W. A., Jr & Hunter, L. E. KaBOB: ontology-based semantic integration of biomedical databases. *BMC Bioinformatics* **16**, 126 (2015).
8. Callahan, T. J., Tripodi, I. J., Pielke-Lombardo, H. & Hunter, L. E. Knowledge-Based Biomedical Data Science. *Annu. Rev. Biomed. Data Sci.* (2020)
doi:10.1146/annurev-biodatasci-010820-091627.
9. Vidal, M., Cusick, M. E. & Barabási, A.-L. Interactome networks and human disease. *Cell* **144**, 986–998 (2011).
10. Crick, F. Central dogma of molecular biology. *Nature* **227**, 561–563 (1970).
11. Berners-Lee, T. Linked Data. *Up to Design Issues*
<http://www.w3.org/DesignIssues/LinkedData.html> (2009).
12. Nicholson, D. N. & Greene, C. S. Constructing knowledge graphs and their biomedical applications. *Comput. Struct. Biotechnol. J.* **18**, 1414–1428 (2020).
13. Ehrlinger, L. & Wöß, W. Towards a Definition of Knowledge Graphs. *SEMANTiCS (Posters, Demos, SuCCESS)* **48**, (2016).
14. Hogan, A. *et al.* Knowledge Graphs. in *ACM Computing Surveys (Csur)* vol. 54 1–37 (2021).
15. Ji, S., Pan, S., Cambria, E., Marttinen, P. & Yu, P. S. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Trans Neural Netw Learn Syst* **33**, 494–514 (2021).
16. Nelson, C. A., Butte, A. J. & Baranzini, S. E. Integrating biomedical research and electronic health records to create knowledge-based biologically meaningful machine-readable embeddings. *Nat. Commun.* **10**, 3045 (2019).
17. Wood, E. C. *et al.* RTX-KG2: a system for building a semantically standardized knowledge graph for translational biomedicine. *BMC Bioinformatics* **23**, 400 (2022).
18. Stear, B. J. *et al.* Petagraph: A large-scale unifying knowledge graph framework for integrating biomolecular and biomedical data. *bioRxiv* (2023)
doi:10.1101/2023.02.11.528088.
19. Belleau, F., Nolin, M.-A., Tourigny, N., Rigault, P. & Morissette, J. Bio2RDF: towards a mashup to build bioinformatics knowledge systems. *J. Biomed. Inform.* **41**, 706–716 (2008).
20. Himmelstein, D. S. *et al.* Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *Elife* **6**, (2017).
21. Dipper. *GitHub* <https://github.com/monarch-initiative/dipper> (2014).
22. The Knowledge Graph Exchange (KGX). *GitHub* <https://github.com/biolink/kgx> (2018).
23. Chung, M.-H., Zhou, J., Pang, X., Tao, Y. & Zhang, J. BioKDE: A deep learning powered search engine and biomedical knowledge discovery platform. in *BioCreative VII Challenge*

- Evaluation Workshop, Virtual workshop* 254–259 (2021).
24. Reese, J. T. *et al.* KG-COVID-19: A Framework to Produce Customized Knowledge Graphs for COVID-19 Response. *Patterns (N Y)* **2**, 100155 (2021).
 25. Chandak, P., Huang, K. & Zitnik, M. Building a Knowledge Graph to Enable Precision Medicine. *Scientific Data* **10**, 67 (2023).
 26. Pratt, D. *et al.* NDEx, the Network Data Exchange. *Cell Syst* **1**, 302–305 (2015).
 27. Caufield, J. H. *et al.* KG-Hub - Building and Exchanging Biological Knowledge Graphs. *Bioinformatics* (2023) doi:10.1093/bioinformatics/btad418.
 28. Santos, A. *et al.* Clinical Knowledge Graph Integrates Proteomics Data into Clinical Decision-Making. *Nat Biotechnol* **40**, 692–702 (2022).
 29. Lobentzner, S. *et al.* Democratising Knowledge Representation with BioCypher. *Nat Biotechnol* (2023) doi:10.1038/s41587-023-01848-y.
 30. Zachary, W. W. An Information Flow Model for Conflict and Fission in Small Groups. *J. Anthropol. Res.* **33**, 452–473 (1977).
 31. Knowledge graphs. *DBpedia Association* <https://www.dbpedia.org/resources/knowledge-graphs/> (2023).
 32. Breit, A., Ott, S., Agibetov, A. & Samwald, M. OpenBioLink: a benchmarking framework for large-scale biomedical link prediction. *Bioinformatics* **36**, 4097–4098 (2020).
 33. Piñero, J. *et al.* DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic Acids Research* **45**, D833–D839 (2017).
 34. Mungall, C. J. *et al.* The Monarch Initiative: an integrative data and analytic platform connecting phenotypes to genotypes across species. *Nucleic Acids Res.* **45**, D712–D722 (2017).
 35. RDF Working Group. RDF - semantic web standards. *w3.org* <https://www.w3.org/RDF/> (2014).
 36. RDF Working Group. RDF 1.1 semantics. *W3C Recommendation* <https://www.w3.org/TR/rdf11-mt/> (2014).
 37. Vettrivel, V. Knowledge graphs: RDF or property graphs, which one should you pick? *Wisecube.ai* <https://www.wisecube.ai/blog/knowledge-graphs-rdf-or-property-graphs-which-one-should-you-pick/> (2022).
 38. The OWL Working Group. OWL Web Ontology Language Overview. *W3C Recommendation* <https://www.w3.org/TR/owl-features/> (2004).
 39. Krötzsch, M., Simancik, F. & Horrocks, I. A Description Logic Primer. *arXiv [cs.AI]* (2012).
 40. Lam, H. Y. K., Marenco, L., Shepherd, G. M., Miller, P. L. & Cheung, K.-H. Using web ontology language to integrate heterogeneous databases in the neurosciences. *AMIA Annu. Symp. Proc.* 464–468 (2006).
 41. Callahan, T. J. *et al.* OWL-NETS: Transforming OWL Representations for Improved Network Inference. in *Biocomputing* 133–144 (WORLD SCIENTIFIC, 2018).
 42. Wilkinson, M. D. *et al.* The FAIR Guiding Principles for scientific data management and stewardship. *Sci Data* **3**, 160018 (2016).
 43. Bio2BEL. *GitHub* <https://github.com/bio2bel> (2017).
 44. Bio2RDF Consortium. Bio2RDF. *GitHub* <https://github.com/bio2rdf> (2011).
 45. Bio4j. *GitHub* <https://github.com/bio4j/bio4j> (2011).
 46. BioGrakn Knowledge Graph. *GitHub* <https://github.com/vaticle/biograkn> (2018).
 47. Clinical Knowledge Graph (CKG). *GitHub* <https://github.com/MannLabs/CKG> (2019).
 48. COVID-19-Community. *GitHub* <https://github.com/covid-19-net/covid-19-community> (2020).

49. Hetionet. *GitHub* <https://github.com/hetio/hetionet> (2016).
50. IASIS Open Data Graph. *GitHub* <https://github.com/tasosnent/Biomedical-Knowledge-Integration> (2018).
51. KG-COVID-19. *GitHub* <https://github.com/Knowledge-Graph-Hub/kg-covid-19> (2020).
52. KaBOB (Knowledge Base Of Biomedicine). *GitHub* <https://github.com/UCDenver-ccp/kabob> (2015).
53. KGTK: Knowledge Graph Toolkit. *GitHub* <https://github.com/usc-isi-i2/kgtk> (2020).
54. ProNet. *GitHub* <https://github.com/cran/ProNet> (2015).
55. Futia, G. semi: A SEmantic Modeling machIne. *GitHub* <https://github.com/giuseppfutia/semi> (2018).
56. Hastings, J. *et al.* ChEBI in 2016: Improved services and an expanding collection of metabolites. *Nucleic Acids Res.* **44**, D1214–9 (2016).
57. Natale, D. A. *et al.* The Protein Ontology: a structured representation of protein forms and complexes. *Nucleic Acids Res.* **39**, D539–45 (2011).
58. Smith, B. *et al.* Relations in biomedical ontologies. *Genome Biol.* **6**, R46 (2005).
59. Eilbeck, K. *et al.* The Sequence Ontology: a tool for the unification of genome annotations. *Genome Biol.* **6**, R44 (2005).
60. Networkx. *density*. <https://networkx.org/documentation/stable/reference/generated/networkx.classes.function.density.html>.
61. Rebele, T. *et al.* YAGO: A Multilingual Knowledge Base from Wikipedia, Wordnet, and Geonames. in *The Semantic Web – ISWC 2016* 177–185 (Springer International Publishing, 2016).
62. Auer, S. *et al.* DBpedia: A Nucleus for a Web of Open Data. in *The Semantic Web* 722–735 (Springer Berlin Heidelberg, 2007).
63. Vrandečić, D. Wikidata: a new platform for collaborative data collection. in *Proceedings of the 21st International Conference on World Wide Web* 1063–1064 (Association for Computing Machinery, 2012).
64. Tiddi, I. & Schlobach, S. Knowledge graphs as tools for explainable machine learning: A survey. *Artif. Intell.* **302**, 103627 (2022).
65. Tripodi, I. J. *et al.* Applying knowledge-driven mechanistic inference to toxicogenomics. *Toxicology in Vitro* **66**, 104877 (2020).
66. Joslyn, C. A. *et al.* Hypernetwork Science: From Multidimensional Networks to Computational Topology. in *International conference on complex systems* (ed. Cham: Springer International Publishing) 377–392 (2020).
67. Callahan, T. J., Hunter, L. E. & Kahn, M. G. *Leveraging a Neural-Symbolic Representation of Biomedical Knowledge to Improve Pediatric Subphenotyping*. (2021). doi:10.5281/zenodo.5746187.
68. Malec, S. A. *et al.* Causal feature selection using a knowledge graph combining structured knowledge from the biomedical literature and ontologies: A use case studying depression as a risk factor for Alzheimer’s disease. *J. Biomed. Inform.* **142**, 104368 (2023).
69. Taneja, S. B. *et al.* Developing a Knowledge Graph for Pharmacokinetic Natural Product-Drug Interactions. *J. Biomed. Inform.* **140**, 104341 (2023).
70. Cavalleri, E. *et al.* A Meta-Graph for the Construction of RNA-KG. in *10th International Work-Conference Bioinformatics and Biomedical Engineering, LNCS vol. 13919* vol. 13919 165–180 (2023).
71. RNA-KG. *GitHub* <https://github.com/AnacletoLAB/RNA-KG> (2023).
72. Santangelo, B. MGMLink. *GitHub* <https://github.com/bsantan/MGMLink> (2022).

73. Cappelletti, L. *et al.* GRAPE for fast and scalable graph processing and random-walk-based embedding. *Nature Computational Science* **3**, 552–568 (2023).
74. Valentini, G. *et al.* Het-node2vec: second order random walk sampling for heterogeneous multigraphs embedding. *arXiv [cs.LG]* (2021).
75. Callahan, T. J. *et al.* Knowledge-Driven Mechanistic Enrichment of the Preeclampsia Ignorome. in *Biocomputing* vol. 28 371–382 (2023).
76. HuBMAP Consortium. The human body at cellular resolution: the NIH Human Biomolecular Atlas Program. *Nature* **574**, 187–192 (2019).
77. Reitz, K. M., Hall, D. E., Shinall, M. C., Jr, Shireman, P. K. & Silverstein, J. C. Using the Unified Medical Language System to expand the Operative Stress Score - first use case. *J. Surg. Res.* **268**, 552–561 (2021).
78. UMLS-Graph: UMLS Graph database for semantic queries. *GitHub* <https://github.com/dbmi-pitt/UMLS-Graph> (2018).
79. ontology-api: The HuBMAP Ontology Service. *GitHub* <https://github.com/hubmapconsortium/ontology-api> (2020).
80. SenNet Consortium. NIH SenNet Consortium to map senescent cells throughout the human lifespan to understand physiological health. *Nat Aging* **2**, 1090–1100 (2022).
81. Santangelo, B. E., Gillenwater, L. A., Salem, N. M. & Hunter, L. E. Molecular cartooning with knowledge graphs. *Front Bioinform* **2**, 1054578 (2022).
82. Cartoomics. *GitHub* <https://github.com/UCDenver-ccp/Cartoomics> (2022).
83. Hoyt, C. T. *et al.* Unifying the identification of biomedical entities with the Bioregistry. *Sci Data* **9**, 714 (2022).
84. 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).
85. Harry Caufield, J. *et al.* KG-Hub -- Building and Exchanging Biological Knowledge Graphs. *Bioinformatics* **btad418**, (2023).
86. OWL Collaboration. owltools. *GitHub* <https://github.com/owlcollab/owltools> (2020).
87. Jackson, R. C. *et al.* ROBOT: A Tool for Automating Ontology Workflows. *BMC Bioinformatics* **20**, 407 (2019).
88. Köhler, S. *et al.* The Human Phenotype Ontology in 2021. *Nucleic Acids Res.* **49**, D1207–D1217 (2021).
89. Ashburner, M. *et al.* Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* **25**, 25–29 (2000).
90. Davis, A. P. *et al.* Comparative Toxicogenomics Database (CTD): update 2021. *Nucleic Acids Res.* **49**, D1138–D1143 (2021).
91. UniProt Consortium. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Res.* **47**, D506–D515 (2019).
92. Szklarczyk, D. *et al.* STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Res.* **47**, D607–D613 (2018).
93. Uhlén, M. *et al.* Proteomics. Tissue-based map of the human proteome. *Science* **347**, 1260419 (2015).
94. GTEx Consortium. The Genotype-Tissue Expression (GTEx) project. *Nat. Genet.* **45**, 580–585 (2013).
95. Shefchek, K. A. *et al.* The Monarch Initiative in 2019: an integrative data and analytic platform connecting phenotypes to genotypes across species. *Nucleic Acids Res.* **48**, D704–D715 (2020).
96. Yates, B. *et al.* Genenames.org: the HGNC and VGNC resources in 2017. *Nucleic Acids Res.*

- 45**, D619–D625 (2017).
97. Maglott, D., Ostell, J., Pruitt, K. D. & Tatusova, T. Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Res.* **33**, D54–8 (2005).
 98. Zerbino, D. R. *et al.* Ensembl 2018. *Nucleic Acids Res.* **46**, D754–D761 (2018).
 99. Mungall, C. J., Torniai, C., Gkoutos, G. V., Lewis, S. E. & Haendel, M. A. Uberon, an integrative multi-species anatomy ontology. *Genome Biol.* **13**, R5 (2012).
 100. Landrum, M. J. *et al.* ClinVar: improving access to variant interpretations and supporting evidence. *Nucleic Acids Res.* **46**, D1062–D1067 (2018).
 101. Callahan, T. J. Data Preparation Jupyter Notebook. *PheKnowLator* https://github.com/callahantiff/PheKnowLator/blob/master/notebooks/Data_Preparation.ipynb (2021).
 102. Amith, M., He, Z., Bian, J., Lossio-Ventura, J. A. & Tao, C. Assessing the practice of biomedical ontology evaluation: Gaps and opportunities. *J. Biomed. Inform.* **80**, 1–13 (2018).
 103. Vrandečić, D. Ontology Evaluation. in *Handbook on Ontologies* (eds. Staab, S. & Studer, R.) 293–313 (Springer Berlin Heidelberg, 2009).
 104. Gómez-Pérez, A. Ontology Evaluation. in *Handbook on Ontologies* (eds. Staab, S. & Studer, R.) 251–273 (Springer Berlin Heidelberg, 2004).
 105. Callahan, T. J. *Ontology Cleaning Jupyter Notebook*. (Github, 2021).
 106. Callahan, T. J. *et al.* *Adapting the Harmonized Data Quality Framework for Ontology Quality Assessment*. (2022). doi:10.5281/zenodo.6941289.
 107. Callahan, T. J. PheKnowLator Human Disease Knowledge Graphs - ontology_cleaning_report.txt. (2021) doi:10.5281/zenodo.7026644.
 108. Hoehndorf, R., Schofield, P. N. & Gkoutos, G. V. The role of ontologies in biological and biomedical research: a functional perspective. *Brief. Bioinform.* **16**, 1069–1080 (2015).
 109. Correia, F. LOGICAL GROUNDS. *Rev. Symb. Log.* **7**, 31–59 (2014).
 110. Baader, F., Calvanese, D., McGuinness, D., Patel-Schneider, P. & Nardi, D. *The Description Logic Handbook: Theory, Implementation and Applications*. (Cambridge University Press, 2003).
 111. Bergman, M. The fundamental importance of keeping an ABox and TBox split. *AI3: Adaptive Information* <https://www.mkbergman.com/489/ontology-best-practices-for-data-driven-applications-part-2/> (2009).
 112. Thessen, A. E. *et al.* Transforming the study of organisms: Phenomic data models and knowledge bases. *PLoS Comput. Biol.* **16**, e1008376 (2020).
 113. Callahan, T. J. OWL-NETS v2.0. *GitHub* <https://github.com/callahantiff/PheKnowLator/wiki/OWL-NETS-2.0> (2021).
 114. Callahan, T. J. PheKnowLator Human Disease Knowledge Graphs - downloaded_build_metadata.txt. (2021) doi:10.5281/zenodo.7026640.
 115. Callahan, T. J. PheKnowLator Human Disease Knowledge Graphs -- Class-based Knowledge Model with Standard Relations (Phases 1-2 Build Log). (v2.1.0_01MAY2021) doi:10.5281/zenodo.7029958.
 116. Callahan, T. J. PheKnowLator Human Disease Knowledge Graphs -- Output File Information. (2022) doi:10.5281/zenodo.7051238.
 117. Xiang, Z. *et al.* VIOLIN: vaccine investigation and online information network. *Nucleic Acids Res.* **36**, D923–8 (2008).
 118. He, Y. *et al.* Updates on the web-based VIOLIN vaccine database and analysis system. *Nucleic Acids Res.* **42**, D1124–32 (2014).
 119. Bard, J., Rhee, S. Y. & Ashburner, M. An ontology for cell types. *Genome Biol.* **6**, R21 (2005).

120. Sarntivijai, S. *et al.* CLO: The cell line ontology. *J. Biomed. Semantics* **5**, 37 (2014).
121. The Gene Ontology Consortium. The Gene Ontology Resource: 20 years and still GOing strong. *Nucleic Acids Res.* **47**, D330–D338 (2019).
122. Vasilevsky, N. A. *et al.* Mondo: Unifying diseases for the world, by the world. *medRxiv* (2022) doi:10.1101/2022.04.13.22273750.
123. Petri, V. *et al.* The pathway ontology - updates and applications. *J. Biomed. Semantics* **5**, 7 (2014).
124. Tsitsulin, A. deepwalk-c: DeepWalk implementation in C++. *GitHub* <https://github.com/xgfs/deepwalk-c> (2017).
125. Callahan, T. J. OWL-NETS Example Application Jupyter Notebook. *GitHub* https://github.com/callahantiff/PheKnowLator/blob/master/notebooks/OWLNETS_Example_Application.ipynb (2022).
126. Callahan, T. J. RDF Graph Processing Example Jupyter Notebook. *GitHub* https://github.com/callahantiff/PheKnowLator/blob/master/notebooks/RDF_Graph_Processing_Example.ipynb (2022).
127. Callahan, T. J. Entity Search Jupyter Notebook. *GitHub* https://github.com/callahantiff/PheKnowLator/blob/master/notebooks/tutorials/entity_search/Entity_Search.ipynb.
128. Database Center for Life Science. sparql-proxy. *GitHub* <https://github.com/dbcls/sparql-proxy> (2016).
129. Blazegraph Database. *Blazegraph* <https://blazegraph.com/> (2015).
130. Features. *GitHub Actions* <https://github.com/features/actions> (2022).
131. Vertex AI. *Google Cloud* <https://cloud.google.com/ai-platform>.
132. Container registry. *Google Cloud* <https://cloud.google.com/container-registry>.
133. Cook, J. *Docker for Data Science*. (Apress Berkeley, CA, 2017).
134. Preston-Werner, T. Semantic Versioning 2.0.0. *Semantic Versioning* <https://semver.org/>.
135. The Apache Software Foundation. *APACHE LICENSE, VERSION 2.0*. <https://www.apache.org/licenses/LICENSE-2.0> (2004).
136. Babar, M. A., Zhu, L. & Jeffery, R. A framework for classifying and comparing software architecture evaluation methods. in *2004 Australian Software Engineering Conference. Proceedings* 309–318 (2004).
137. Callahan, T. J., Baumgartner, W. A. & Hunter, L. E. Biomedical KG Construction Survey. (2021) doi:10.5281/ZENODO.5790040.
138. Callahan, T. J. GitHub API Repository Search (Python 3.6.2). *GitHub* <https://gist.github.com/callahantiff/0ae1c00df9bec7228be3f6bda5466d73> (2020).
139. Bastian, M., Heymann, S. & Jacomy, M. Gephi: An Open Source Software for Exploring and Manipulating Networks. *ICWSM* **3**, 361–362 (2009).
140. Martin, S., Michael Brown, W., Klavans, R. & Boyack, K. W. OpenOrd: an open-source toolbox for large graph layout. in *Visualization and Data Analysis 2011* vol. 7868 786806 (International Society for Optics and Photonics, 2011).
141. Callahan, T. J. *Overview of the PheKnowLator Ecosystem*. (2022). doi:10.5281/zenodo.6998816.

Acknowledgements

This work was supported by funding from the National Library of Medicine (T15LM009451 and T15LM007079) to TJC, (K99LM013367) to SAM, (R01LM013400 and 5R01LM008111-16) to LEH, and (R01LM006910) GH. This work was also supported by funding from the Director, Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract (DE-AC02-05CH11231) to JR, the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR), Award (URF/1/4355-01-01 and URF/1/5041-01-01) to RH, the NIH Common Fund (CFDE OT2OD030545, HuBMAP OT2OD033759, and SenNet U24CA268108) to JCS, and the Defense Advanced Research Projects Agency (DARPA) Young Faculty Award (W911NF-20-1-0255) and the DARPA Automating Scientific Knowledge Extraction and Modeling program (HR00112220036) to CTH. The authors would like to thank the OHDSI community, especially Adam Black as well as members of Dr. Hunter's lab at the University of Colorado Anschutz Medical Campus, specifically Mayla Boguslav and Harrison Pielke-Lombardo for testing different builds and helping conceive and pilot test tutorials to demonstrate different PheKnowLator use cases. The author's would also like to thank GitHub users nomisto, Bancherd-DeLong, Bsantan, and ablack3, who identified and helped troubleshoot bugs through the PheKnowLator GitHub.

Author Contributions

MGK, WAB, and LEH served as primary supervisors of this work. TJC, BAW, ALS, IJT, RH, and ALS conceived and helped develop the analyses performed in this work. TJC and WAB developed the PheKnowLator ecosystem. ALS, IJT, and JMW provided insight into the development of documentation for the GitHub site. ALS, BS, CJM, CTH, FM, GH, JCS, JH, JMW, JR, MB, NAM, NAV, PBR, PNR, RDB, RH, and TDB provided domain expertise and/or commented on the PheKnowLator ecosystem, data sources, or other important resources used in its development. BS, EIC, EmC, GV, LC, LG, MM, RB, SAM, SBT, and TF evaluated PheKnowLator builds and provided feedback on the resulting KGs. TJC drafted the manuscript and all authors reviewed the manuscript and provided feedback. All authors read and approved the final version of the manuscript.

Competing Interests

The authors declare no competing interests.

Table 1. Open Source Knowledge Graph Construction Methods.

Method	GitHub repository
Bio2BEL	https://github.com/bio2bel/
Bio2RDF	https://github.com/bio2rdf
Bio4J	https://github.com/bio4j/bio4j
BioGrakn	https://github.com/graknlabs/biograkn
Clinical Knowledge Graph (CKG)	https://github.com/MannLabs/CKG
COVID-19-Community	https://github.com/covid-19-net/covid-19-community
Dipper	https://github.com/monarch-initiative/dipper
Hetionet	https://github.com/hetio/hetionet
iASiS Open Data Graph	https://github.com/tasosnent/Biomedical-Knowledge-Integration
KG-COVID-19	https://github.com/Knowledge-Graph-Hub/kg-covid-19
Knowledge Base Of Biomedicine (KaBOB)	https://github.com/UCDenver-ccp/kabob/tree/bg-integration
Knowledge Graph Exchange (KGX)	https://github.com/NCATS-Tangerine/kgx
Knowledge Graph Toolkit (KGTK)	https://github.com/usc-isi-i2/kgtk/
ProNet	https://github.com/cran/ProNet
SEmantic Modeling machine (SEMi)	https://github.com/giuseppfutia/semi

Table 2. PKT Human Disease Knowledge Graph Primary Node Types.

Node	Universal Resource Identifier
Anatomical Entities	http://purl.obolibrary.org/obo/UBERON
Biological Processes	http://purl.obolibrary.org/obo/GO
Catalysts	http://purl.obolibrary.org/obo/CHEBI
Cells	http://purl.obolibrary.org/obo/CL
Cell Lines	http://purl.obolibrary.org/obo/CLO
Cellular Components	http://purl.obolibrary.org/obo/GO
Chemicals	http://purl.obolibrary.org/obo/CHEBI
Cofactors	http://purl.obolibrary.org/obo/CHEBI
Diseases	http://purl.obolibrary.org/obo/MONDO
Genes	http://www.ncbi.nlm.nih.gov/gene/
Molecular Functions	http://purl.obolibrary.org/obo/GO
Pathways ^a	http://purl.obolibrary.org/obo/PW https://reactome.org/content/detail/R-HSA-
Phenotypes	http://purl.obolibrary.org/obo/HP
Proteins	http://purl.obolibrary.org/obo/PRO
Sequences ^b	http://purl.obolibrary.org/obo/SO
Transcripts	https://uswest.ensembl.org/Homo_sapiens/Transcript/Summary?t=ENST
Vaccines ^b	http://purl.obolibrary.org/obo/VO
Variants	https://www.ncbi.nlm.nih.gov/snp/rs

Note: The node types listed above apply to the PKT Human Disease KG v2.1.0. The node types listed above do not include all of the classes that exist in each Open Biological and Biomedical Ontology (OBO) Foundry ontology. The Cell Ontology is included with the extended version of Uberon.

^aTwo URIs are shown for pathways as the OBO Found ontology is the core ontology used to connect Reactome entities to the core set of OBO Foundry ontologies.

^bOBO node type. Includes all of the classes that are contained in the ontology even though they are not all explicitly listed here.

Acronyms: CL (Cell ontology); CLO (Cell Line Ontology); CHEBI (Chemical Entities of Biological Interest); GO (Gene Ontology); HPO (Human Phenotype Ontology); MONDO (Mondo Disease Ontology); PKT (PheKnowlator); PRO (Protein Ontology); PW (Pathway Ontology); SO (Sequence Ontology); VO (Vaccine Ontology); UBERON (Uber-Anatomy Ontology).

Table 3. PKT Human Disease Knowledge Graph Primary Relations and Edge Types.

Relations	Edge Types
participates in (RO_0000056) has participant (RO_0000057)	chemical-pathway; gene-pathway; protein-biological process; protein-pathway
has function (RO_0000085) function of (RO_0000079)	pathway-molecular function; protein-molecular function
located in (RO_0001025) location of (RO_0001015)	protein-anatomy; protein-cell ^a ; protein-cellular component; transcript-anatomy; transcript-cell ^a
has component (RO_0002180) ^b	pathway-cellular component
has phenotype (RO_0002200) phenotype of (RO_0002201)	disease-phenotype
has gene product (RO_0002205) gene product of (RO_0002204)	gene-protein
interacts with (RO_0002434) ^c	chemical-gene; chemical-protein
genetically interacts with (RO_0002435) ^c	gene-gene
molecularly interacts with (RO_0002436) ^c	chemical-biological process; chemical-cellular component; chemical-molecular function; protein-catalyst; protein-cofactor; protein-protein
transcribed to (RO_0002511) transcribed from (RO_0002510)	gene-transcript
ribosomally translates to (RO_0002513) ribosomal Translation of (RO_0002512)	transcript-protein
causally influences (RO_0002566) causally influenced by (RO_0002559)	variant-gene
is substance that treats (RO_0002606) is treated by substance (RO_0002302)	chemical-disease; chemical-phenotype
causes or contributes to condition (RO_0003302) ^b	gene-disease; gene-phenotype; variant-disease; variant-phenotype
realized in response to (RO_0009501) ^b	biological process-pathway

Note: The primary relations and edge types listed above apply to the PKT Human Disease KG v2.1.0. These relations are added to the core set of Open Biological and Biomedical Ontology Foundry ontologies.

^aThe word "cell" above is used to represent cell lines from the Cell Line Ontology and cell types from the Cell Ontology.

^bRelation Ontology concepts that do not have an inverse.

^cRelations with symmetrical inverse relations.

Acronyms: PKT (PheKnowlator).

Table 4. Ontology Statistics Pre- and Post-Data Quality Assessment.

Ontology	Before Cleaning		After Cleaning	
	Classes	Triples	Classes	Triples
Cell Line Ontology	111,712	1,387,096	111,696	1,422,153
Chemical Entities of Biological Interest	156,098	5,264,571	137,592	5,190,485
Gene Ontology	62,237	1,425,434	55,807	1,343,218
Human Phenotype Ontology	38,843	884,999	38,530	885,379
Mondo Disease Ontology	55,478	2,313,343	52,937	2,277,425
Protein Ontology ^a	148,243	2,079,356	148,243	2,079,356
Pathway Ontology	2,642	35,291	2,600	34,901
Relation Ontology	116	7,970	115	7,873
Sequence Ontology	2,910	44,655	2,569	41,980
Uber-Anatomy Ontology ^b	28,738	752,291	27,170	734,768
Vaccine Ontology	7,089	86,454	7,085	89,764
Core OBO Foundry ontologies (merged) ^c	548,947	13,746,883	545,259	13,748,009

Note: The numbers for the ontologies are calculated using the versions of the ontologies that include all imported ontologies referenced by the primary ontology. This means that the counts of classes include all OWL classes used for logical definitions, not only those that are explicitly part of the primary ontology's namespace.

^aThe Protein Ontology version references the human subset created for the PheKnowLator ecosystem.

^bThe extended version of the Uber-Anatomy Ontology contains the Cell Ontology.

^cConsistency was evaluated using the ELK reasoner. The reasoner was only applied to individual ontologies.

Table 5. PKT Human Disease Knowledge Graph Descriptive Statistics by Primary Edge Type.

Edge	Relation	Subjects	Objects	Standard Relations	Inverse Relations
chemical-disease	substance that treats	4,289	4,494	167,681	335,362
chemical-gene ^a	interacts with	462	11,922	16,639	33,278
chemical-biological process ^a	molecularly interacts with	1,338	1,569	287,068	574,136
chemical-cellular component ^a	molecularly interacts with	1,085	226	40,992	81,984
chemical-molecular function ^a	molecularly interacts with	1,105	200	25,385	50,770
chemical-pathway	participates in	2,104	2,213	28,685	57,370
chemical-phenotype	substance that treats	4,053	1,712	107,962	215,924
chemical-protein ^a	interacts with	4,178	6,379	64,991	129,982
disease-phenotype	has phenotype	11,620	9,714	408,702	817,404
gene-disease ^b	causes or contributes to	5,031	4,420	12,717	---
gene-gene ^a	genetically interacts with	247	263	1,668	3,336
gene-pathway	participates in	10,371	1,809	104,906	209,812
gene-phenotype ^b	causes or contributes to	6,780	1,528	23,501	---
gene-protein	has gene product	19,327	19,143	19,534	39,068
gene-transcript	transcribed to	25,529	179,870	182,736	365,472
biological process-pathway ^b	realized in response to	471	665	665	---
pathway-cellular component ^b	has component	11,134	99	15,846	---
pathway-molecular function	has function	2,412	726	2,416	4,832
protein-anatomy	located in	10,747	68	30,682	61,364
protein-catalyst ^a	molecularly interacts with	3,024	3,730	23,629	47,258
protein-cell ^c	located in	10,045	125	73,530	147,060
protein-cofactor ^a	molecularly interacts with	1,584	44	1,961	3,922
protein-biological process	participates in	17,527	12,246	137,812	275,624
protein-cellular component	located in	18,427	1,757	81,602	163,204
protein-molecular function	has function	17,779	4,324	68,633	137,266
protein-pathway	participates in	10,852	2,468	117,182	234,364
protein-protein ^d	molecularly interacts with	14,320	14,230	618,069	---
transcript-anatomy	located in	29,104	102	439,917	879,834
transcript-cell ^c	located in	14,038	127	64,427	128,854
transcript-protein	ribosomally translates to	44,144	19,200	44,147	88,294
variant-disease ^b	causes or contributes to	13,291	3,565	37,861	---
variant-gene	causally influences	121,790	3,236	121,790	243,580
variant-phenotype ^b	causes or contributes to	1,822	371	2,470	---

Please see Table 3 for Relation Ontology for inverse relations and identifiers.

^aSymmetric relations were computationally inferred.

^bThe Relation Ontology does not provide an inverse relation.

^cThe word "cell" above is used to represent cell lines from the Cell Line Ontology and cell types from the Cell Ontology.

^dThe data source already included symmetrical edges.

Acronyms: PKT (PheKnowlator).

Table 6. PheKnowLator Human Disease Knowledge Graph Descriptive Statistics.

Knowledge Model	Relation Strategy	Semantic Abstraction	Edges (triples)	Nodes	Relations	Self-Loops	Average Degree
^a Core OBO Foundry ontologies	N/A	N/A	4,044,658	1,399,756	847	3	2.89
Class-based	Standard Relations	None	25,143,729	8,479,167	847	3	2.97
		Semantic Abstraction Only	4,967,427	743,829	294	445	6.68
		Semantic Abstraction + Harmonization	4,967,429	743,829	293	445	6.68
	Inverse Relations	None	41,116,791	13,803,521	847	3	2.98
		Semantic Abstraction Only	7,629,597	743,829	301	445	10.26
		Semantic Abstraction + Harmonization	7,629,599	743,829	300	445	10.26
Instance-based	Standard Relations	None	21,770,455	8,479,167	847	3	2.57
		Semantic Abstraction Only	4,967,391	743,829	294	409	6.68
		Semantic Abstraction + Harmonization	7,285,496	743,829	293	649	9.79
	Inverse Relations	None	24,432,633	8,479,167	847	3	2.88
		Semantic Abstraction Only	7,629,594	743,829	301	409	10.26
		Semantic Abstraction + Harmonization	9,624,232	743,829	300	650	12.94

Note. Edges and triples are synonymous with respect to the results reported in this table.

^aRelation Strategy and Semantic Abstraction information are not provided as this row of the table reports information on the core set of merged ontologies.

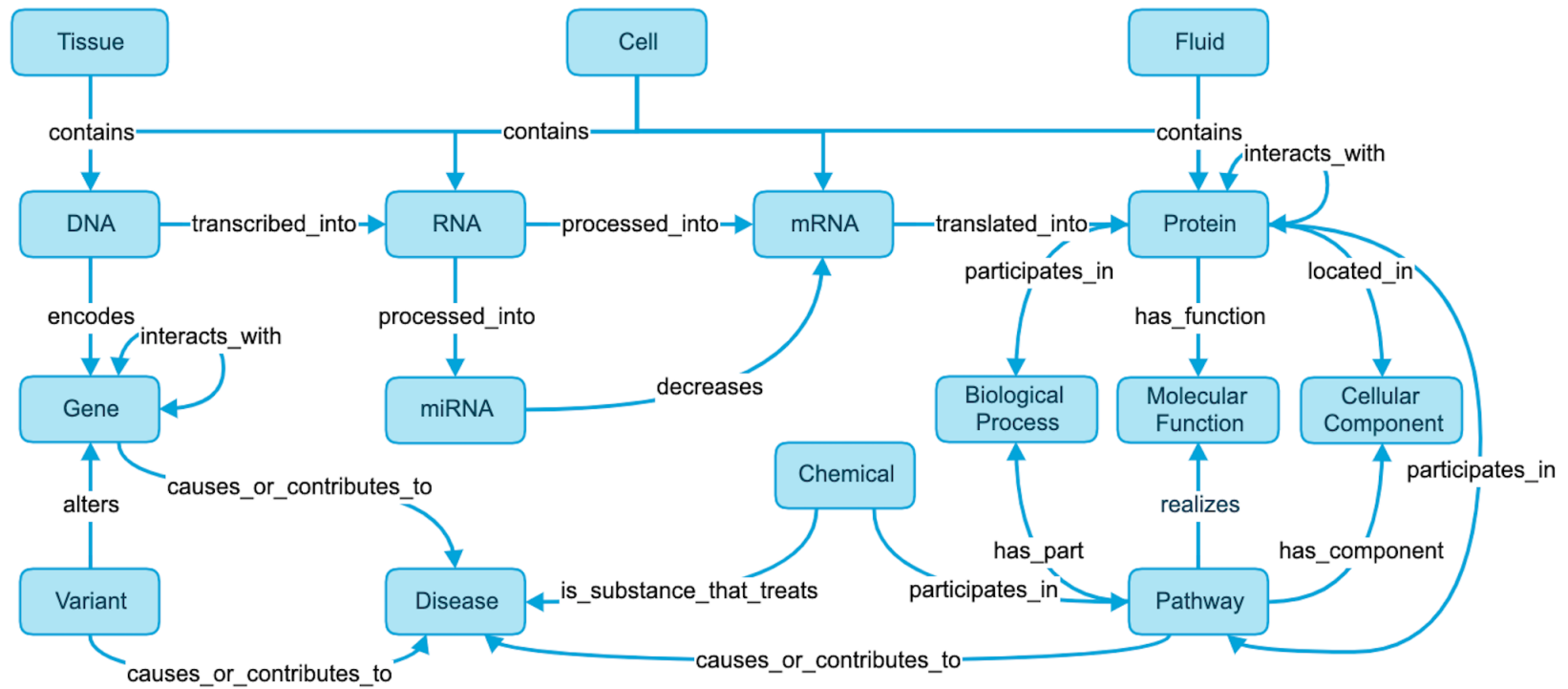


Figure 1. A Knowledge Representation of the Levels of Biological Organization Underlying Human Disease.

This knowledge graph provides a representation of our currently accepted knowledge of the Central Dogma expanded to include pathways, variants, pharmaceutical treatments, and diseases.¹⁰ At a high level this knowledge graph represents anatomical entities like tissues, cells, and bodily fluids containing genomic entities like DNA, RNA, mRNA, and proteins. DNA encodes genes that are processed into mRNA and translated into proteins, which can interact with each other. Genes can also be altered by variants and may cause disease. Finally, proteins also have molecular functions and participate in pathways and biological processes.

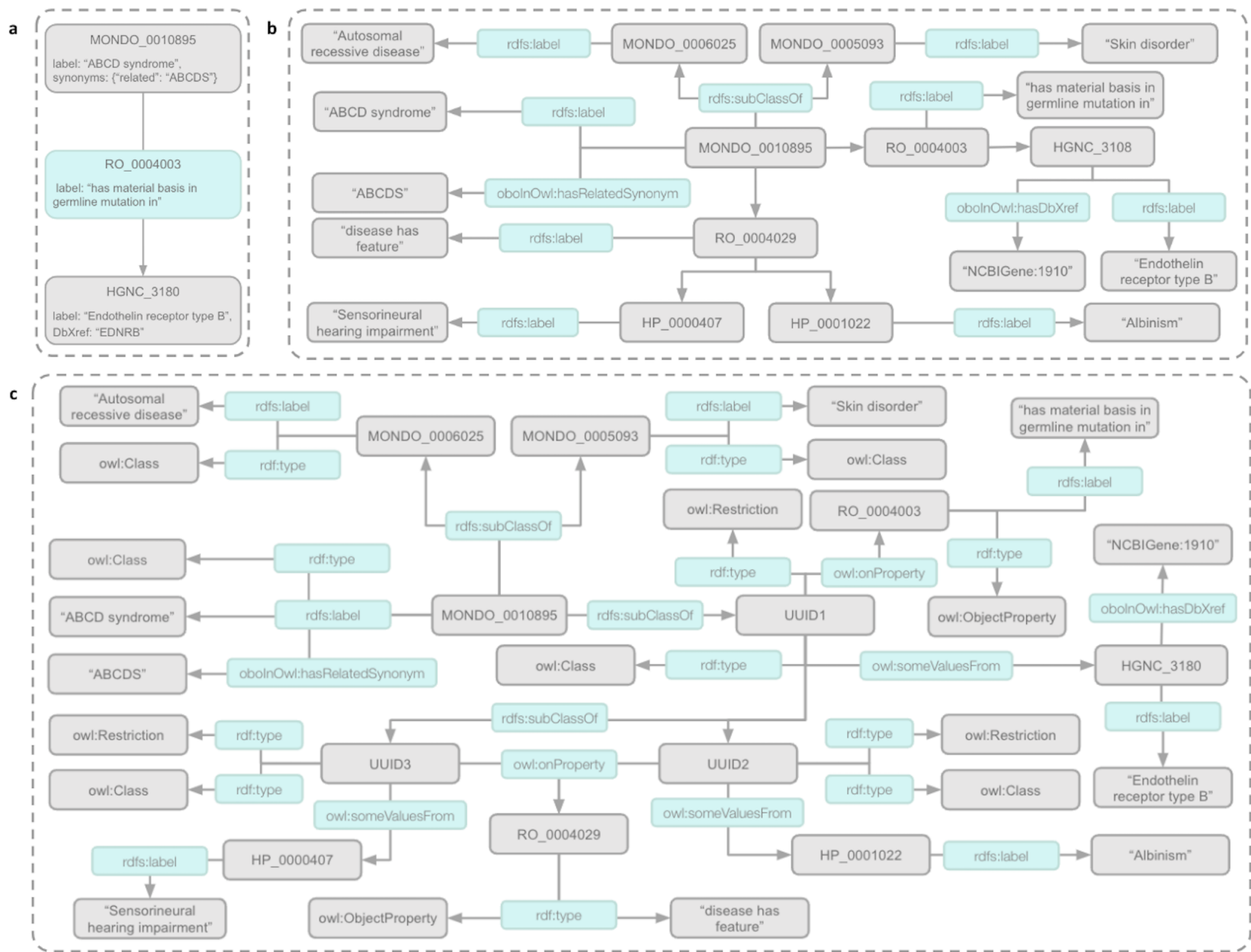


Figure 2. Types of Knowledge Graphs used in the Life Sciences.

This figure provides examples of three types of knowledge graphs that are typically used in the Life Sciences. All knowledge graphs are modeling the Mondo concept ABCD syndrome (*MONDO:0010895*). **(A)** illustrates a simple graph-based representation where two nodes are connected by an edge and nodes and edges are assigned attributes in the form of key-value pairs. **(B)** illustrates a hybrid or property graph-based representation where edges are represented as sets of three nodes (each composed of a subject, predicate, and object) called triples, often based on the RDF/RDFS standards. **(C)** illustrates a complex or OWL-graph-based representation where edges are represented as triples and these representations are augmented with additional OWL expressivities such as domain or range restrictions and description logic. Acronyms: HP (Human Phenotype Ontology); MONDO (Mondo Disease Ontology); OWL (Web Ontology Language); RDF (Resource Description Framework); RDFS (Resource Description Framework Syntax); RO (Relation Ontology).

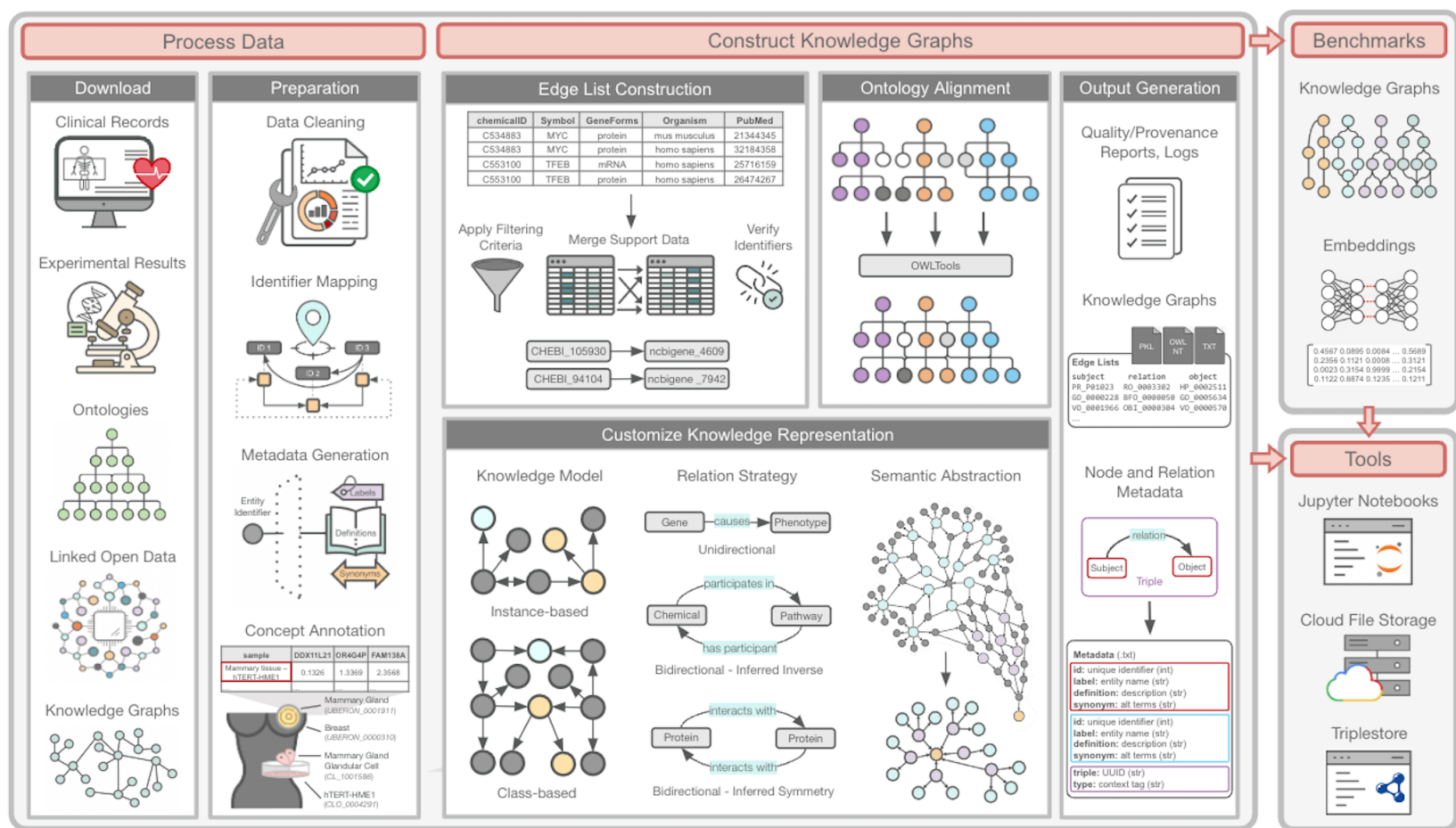


Figure 3. The PheKnowLator Ecosystem.

This figure provides an overview of the PheKnowLator ecosystem.¹⁴¹ The ecosystem consists of three components as indicated by the gray boxes: (1) **Knowledge Graph Construction Resources**, which consist of resources to download and process data and an algorithm to customize the construction of large-scale heterogeneous biomedical knowledge graphs; (2) **Knowledge Graph Benchmarks**, which consist of prebuilt KGs that can be used to systematically assess the effects of different knowledge representations on downstream analyses, workflows, and learning algorithms; and (3) **Knowledge Graph Tools** to use knowledge graphs, cloud-based data storage, APIs, and triplestores. Acronyms: NT (N-Triples file format); OWL (Web Ontology Language); PKL (Python pickle file format); SPARQL (SPARQL Protocol and RDF Query Language).

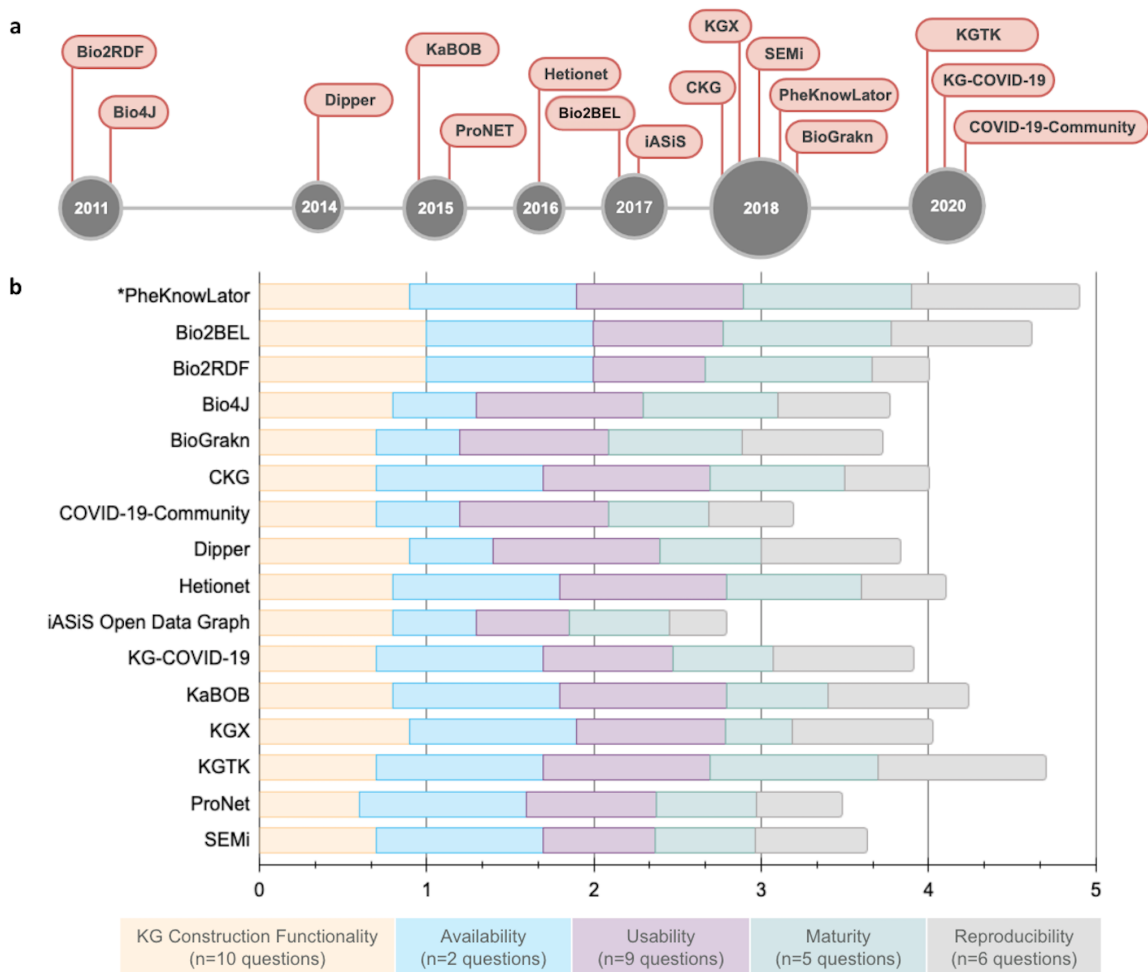


Figure 4. Open-Source Knowledge Graph Construction Methods - Survey Results.

This figure presents the open-source knowledge graph construction methods identified on GitHub and the results of the survey assessment. **(A)** The final set of 16 knowledge graph construction methods surveyed according to the year they were first published on GitHub. **(B)** A chart of the methods evaluated in terms of the different survey categories. The survey was scored out of a total score of five points, which was derived as the sum of the ratios of coverage, each out of one point, for the five categories: KG Construction Functionality (10 questions); Availability (two questions); Usability (nine questions); Maturity (five questions); and Reproducibility (six questions). Acronyms: iASiS, Automated Semantic Integration of Disease-Specific Knowledge; KaBOB, Knowledge Base Of Biomedicine; KG, (Knowledge Graph); KGX (Knowledge Graph Exchange); KGTK (Knowledge Graph Toolkit); SEMi (SEmantic Modeling machine).

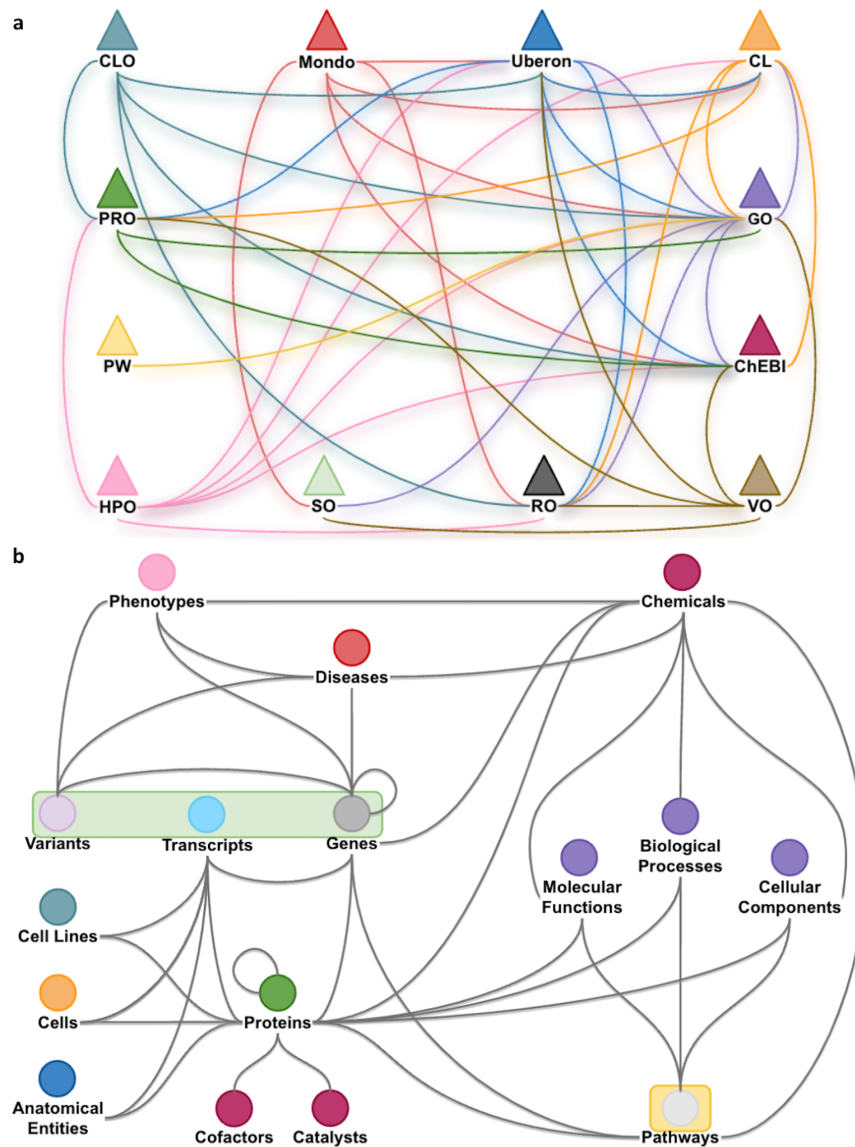


Figure 5. An Overview of the PKT Human Disease Mechanism Knowledge Graph.

This figure provides a high-level overview of the primary node and edge types in the PKT Human Disease Mechanism knowledge graph. (A) illustrates the relationships between the core set of Open Biological and Biomedical Ontology (OBO) Foundry ontologies when including their imported ontologies (as of August 2022). (B) illustrates the edges or triples that are added to the core set of merged ontologies in (A). Shared colors between (A) and (B) represent a single resource. For example, chemicals, cofactors, and catalysts share the same color (maroon) and are part of ChEBI. This is the same for the RO, which is represented in (B) as the black lines between nodes. The green and yellow rectangles indicate data sources that are not from an OBO Foundry ontology and the specific ontology used to integrate them with the core set of ontologies in (A). For example, variant, transcript, and gene data are connected to the core ontology set via the SO. Acronyms: CL (Cell ontology); CLO (Cell Line Ontology); ChEBI (Chemical Entities of Biological Interest); GO (Gene Ontology); HPO (Human Phenotype Ontology); Mondo (Mondo Disease Ontology); PRO (Protein Ontology); PW (Pathway Ontology); SO (Sequence Ontology); VO (Vaccine Ontology); Uberon (Uber-Anatomy Ontology).

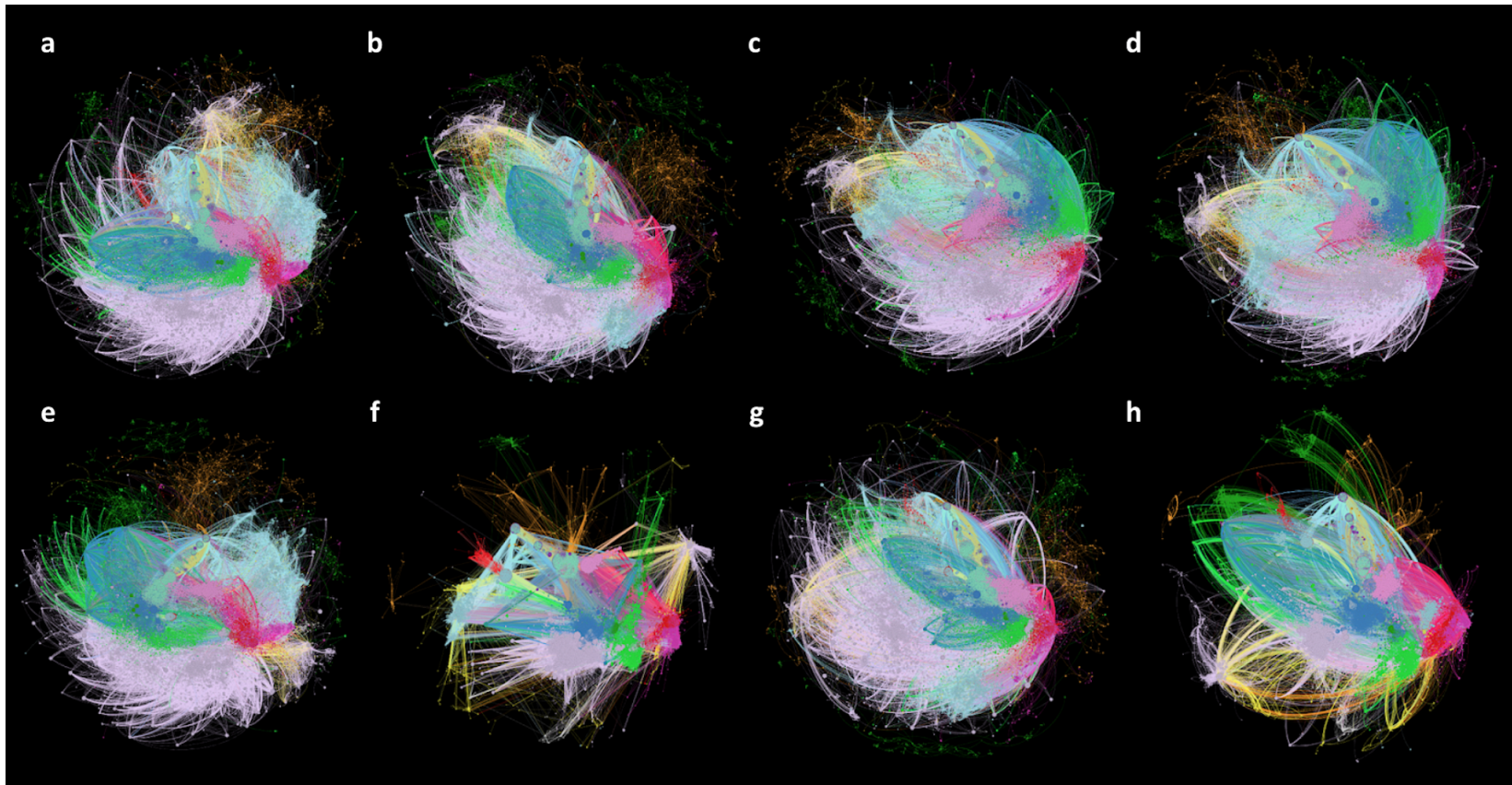


Figure 6. The Impact of Knowledge Model Harmonization on the Semantically Abstracted PKT Human Disease Knowledge Graphs.

The figure visualizes the impact of knowledge model harmonization on the semantically abstracted PKT Human Disease benchmark Knowledge Graphs. The top row of figures (A-D) were built using the class-based knowledge model varying: (A) standard relations without harmonization; (B) standard relations with harmonization; (C) inverse relations without harmonization; (D) inverse relations with harmonization. The bottom row of figures (E-H) were built using the instance-based knowledge model varying: (E) standard relations without harmonization; (F) standard relations with harmonization; (G) inverse relations without harmonization; (H) inverse relations with harmonization. Nodes are colored by type: anatomical entities (light blue), chemical entities (light purple), diseases (red), genes (purple), genomic features (light green), organisms (yellow), pathways (dark green), phenotypes (magenta), proteins (dark blue), molecular sequences (orange), transcripts (turquoise), and variants (light pink).

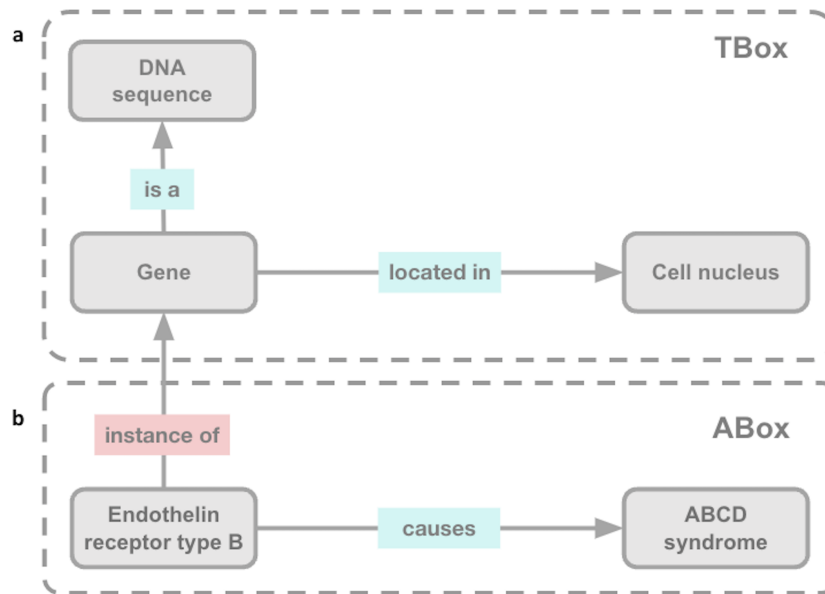


Figure 7. Description Logics Approaches to Knowledge Modeling.

This figure provides a simple example of two approaches for modeling knowledge within a Description Logics architecture. **(A)** The TBox includes classes (i.e., “Gene”, “DNA sequence”, and “Cell nucleus”), properties (i.e., “located in” and “is a”), and the assertions between classes (i.e., “Gene is a DNA sequence” and “Gene located in Cell nucleus”). **(B)** The ABox includes instances of classes (i.e., “Endothelin receptor type B”) represented in the TBox and assertions about those instances (i.e., “Endothelin receptor type B, instance of, Gene” and “Endothelin receptor type B, causes, ABCD syndrome”). Please note that this figure is a simplification and was inspired by Figure 2 from Thessen et al. (2020)¹¹².

An Open-Source Knowledge Graph Ecosystem for the Life Sciences

SUPPLEMENTARY MATERIAL

Tiffany J. Callahan, Ignacio J. Tripodi, Adrienne L. Stefanski, Luca Cappelletti, Sanya B. Taneja, Jordan M. Wyrwa, Elena Casiraghi, Nicolas A. Matentzoglou, Justin Reese, Jonathan C. Silverstein, Charles Tapley Hoyt, Richard D. Boyce, Scott A. Malec, Deepak R. Unni, Marcin P. Joachimiak, Peter N. Robinson, Christopher J. Mungall, Emanuele Cavalleri, Tommaso Fontana, Giorgio Valentini, Marco Mesiti, Lucas A. Gillenwater, Brook Santangelo, Nicole A. Vasilevsky, Robert Hoehndorf, Tellen D. Bennett, Patrick B. Ryan, George Hripcsak, Michael G. Kahn, Michael Bada, William A. Baumgartner Jr, Lawrence E. Hunter

Table of Contents

Supplementary Table 1. Important Definitions.	2
Supplementary Table 2. Acronyms used in the Manuscript.	3
Supplementary Table 3. PheKnowLator Ecosystem Resources.	3
Supplementary Table 4. PheKnowLator Ecosystem Evaluation Resources.	6
Supplementary Table 5. Open-Source Knowledge Graph Construction Methods Survey Criteria.	7
Supplementary Table 6. Open-Source Knowledge Graph Construction Methods.	8
Supplementary Table 7. Open-Source Knowledge Graph Construction Survey - Functionality.	10
Supplementary Table 8. Open-Source Knowledge Graph Construction Survey - Availability.	11
Supplementary Table 9. Open-Source Knowledge Graph Construction Survey - Usability.	12
Supplementary Table 10. Open-Source Knowledge Graph Construction Survey - Maturity.	13
Supplementary Table 11. Open-Source Knowledge Graph Construction Survey - Reproducibility.	14
Supplementary Table 12. PKT Human Disease Knowledge Graph Resources.	15
Supplementary Table 13. Application of Data Quality Checks to OBO Foundry Ontologies.	17
Supplementary Table 14. PheKnowLator Knowledge Modeling Approaches.	18
Supplementary Figure 1. Human Disease Mechanism Graph Knowledge Representation.	19
Supplementary Figure 2. PKT Human Disease Knowledge Graph Construction - Computational Performance.	20
Supplementary Figure 3. The PheKnowLator Ecosystem on FAIR Principles.	21

Supplementary Table 1. Important Definitions.

Concept	Definition
Database	A data source not represented as an ontology, which can include Linked Open Data, data from experiments, clinical data, and existing networks and knowledge graphs.
Edge	Observed connections between nodes. Edges or triples can also be thought of as node-relation-node statements (e.g., geneA - interacts with - geneB).
Graph	An undirected, unweighted network $G(N, L)$, where N is the set of nodes and L is the set of observed edges between these nodes.
Knowledge Graph	A graph-based data structure representing a variety of heterogeneous entities (i.e., nodes) and multiple types of relationships between them and serving as an abstract framework that is able to infer new knowledge to address a variety of applications and use cases.
Knowledge Model	Within the PheKnowLator Ecosystem, there are two types of Knowledge Models that can be used when constructing a knowledge graph: (i) class-based, in which, KGs are constructed using classes, with database entities connected to the core set of merged ontologies as subclasses of existing ontology classes and (ii) instance-based, in which KGs are constructed using instances, with database entities connected to the core set of merged ontologies as instances of existing ontology classes).
Node	Entities or concepts, which are the subject of a knowledge graph. In the biomedical context, nodes usually represent different kinds of biological entities like genes, proteins or diseases.
PKT Human Disease KG	PheKnowLator Ecosystem benchmark knowledge graphs that represent the molecular mechanisms of human disease.
PKT-KG	Phenotype Knowledge TransLator knowledge graph construction algorithm.
Relation	The relationship that connects two nodes in a triple or edge. Relations are used to specify different types of relationships (e.g., interaction, substance that treats) that can exist between a pair of nodes.
Relation Strategy	Within the PheKnowLator Ecosystem, relations can be modeled in two ways when constructing a knowledge graph: (i) Standard Relations (i.e., a unidirectional edge is used to connect a pair of nodes) and (ii) Inverse Relations (i.e., bidirectional edges created by inferring the inverse of relations from ontologies and implicitly symmetric relations like gene-gene interactions).
Semantic Abstraction	Within the PheKnowLator Ecosystem, the OWL-NETS algorithm is used to decode semantically complex OWL-based KGs into KGs that contain biologically meaningful information. Additionally, the Semantic Abstraction parameter, when used with OWL-NETS, includes functionality that can harmonize a KG to a specific kind of Knowledge Model. See the following link for more information: https://github.com/callahantiff/PheKnowLator/wiki/OWL-NETS-2.0 .

Supplementary Table 2. Acronyms used in the Manuscript.

Concept	Definition
API	Application Programming Interfaces
ChEBI	Chemical Entities of Biological Interest Ontology
CL	Cell Ontology
CLO	Cell Line Ontology
FAIR	Findable, Accessible, Interoperable, and Reproducible
GCS	Google Cloud Storage
GB	Gigabyte
GO	Gene Ontology
HPO	Human Phenotype Ontology
KG	Knowledge Graph
Mondo	Mondo Disease Ontology
OBO	Open Biological and Biomedical Ontology
OWL	Web Ontology Language
PheKnowLator	Phenotype Knowledge Translator
PKT	PheKnowLator
PRO	Protein Ontology
PW	Pathway Ontology
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RO	Relation Ontology
SO	Sequence Ontology
SPARQL	SPARQL Protocol and RDF Query Language
Uberon	Uber-Anatomy Ontology
VO	Vaccine Ontology
XML	Extensible Markup Language

Supplementary Table 3. PheKnowLator Ecosystem Resources.

Resource	URL
Ecosystem Component 1: Knowledge Graph Construction Resources	
GitHub	https://github.com/callahantiff/PheKnowLator
PyPI	https://pypi.org/project/pkt-kg/
Docker Container	pkt-kg: https://github.com/callahantiff/PheKnowLator/blob/master/Dockerfile
DockerHub	https://hub.docker.com/repository/docker/callahantiff/pheknowlator
GitHub Actions	https://github.com/callahantiff/PheKnowLator/blob/master/.github/workflows/build-qa.yml
Algorithm Dependencies	https://github.com/callahantiff/PheKnowLator/wiki/Dependencies
Dependency Automation Script	https://github.com/callahantiff/PheKnowLator/blob/master/generates_dependency_documents.py
Testing Suite	https://github.com/callahantiff/PheKnowLator/tree/master/tests
Data Processing Jupyter Notebooks	https://github.com/callahantiff/PheKnowLator/blob/master/notebooks/Data_Preparation.ipynb https://github.com/callahantiff/PheKnowLator/blob/master/notebooks/Ontology_Cleaning.ipynb
Ecosystem Component 2: Knowledge Graph Benchmarks	
Human Disease KG Benchmark Details	https://github.com/callahantiff/PheKnowLator/wiki/Benchmarks-and-Builds
Zenodo Human Disease KG Benchmark Builds Archive	https://zenodo.org/communities/pheknowlator-benchmark-human-disease-kg
GCS Human Disease KG Benchmark Builds Archive	https://console.cloud.google.com/storage/browser/pheknowlator
Human Disease KG Benchmark Builds	https://doi.org/10.5281/zenodo.7030039
<i>Knowledge Graph Build Workflow Scripts</i>	
Build Documentation	https://github.com/callahantiff/PheKnowLator/blob/master/builds
Docker Containers	https://github.com/callahantiff/PheKnowLator/blob/master/builds/Dockerfile.phases12 https://github.com/callahantiff/PheKnowLator/blob/master/builds/Dockerfile.phase3
GitHub Actions	https://github.com/callahantiff/PheKnowLator/blob/master/.github/workflows/kg-build-part1.yml https://github.com/callahantiff/PheKnowLator/blob/master/.github/workflows/kg-build-part2.yml
Build Requirements	https://github.com/callahantiff/PheKnowLator/blob/master/builds/build_requirements.txt
Build Utilities	https://github.com/callahantiff/PheKnowLator/blob/master/builds/build_utilities.py
Build Logging	GCP: https://github.com/callahantiff/PheKnowLator/blob/master/builds/job_monitoring.py pkt-kg: https://github.com/callahantiff/PheKnowLator/blob/master/builds/logging.ini
Phase 1 Build Scripts	https://github.com/callahantiff/PheKnowLator/blob/master/builds/phases1_2_entrypoint.py https://github.com/callahantiff/PheKnowLator/blob/master/builds/build_phase_1.py https://github.com/callahantiff/PheKnowLator/blob/master/builds/data_to_download.txt
Phase 2 Build Scripts	https://github.com/callahantiff/PheKnowLator/blob/master/builds/phases1_2_entrypoint.py https://github.com/callahantiff/PheKnowLator/blob/master/builds/build_phase_2.py https://github.com/callahantiff/PheKnowLator/blob/master/builds/data_preprocessing.py https://github.com/callahantiff/PheKnowLator/blob/master/builds/ontology_cleaning.py
Phase 3 Build Scripts	https://github.com/callahantiff/PheKnowLator/blob/master/builds/build_phase_3.py https://github.com/callahantiff/PheKnowLator/blob/master/builds/phase3_log_daemon.py

Resource	URL
Ecosystem Component 3: Knowledge Graphs Tools	
Zenodo Community	https://zenodo.org/communities/pheknowlator-ecosystem
GCS Bucket	https://console.cloud.google.com/storage/browser/pheknowlator
Jupyter Notebooks	https://github.com/callahantiff/PheKnowLator/blob/master/main.ipynb https://github.com/callahantiff/PheKnowLator/blob/master/notebooks/OWLNETS_Example_Application.ipynb https://github.com/callahantiff/PheKnowLator/blob/master/notebooks/RDF_Graph_Processing_Example.ipynb https://github.com/callahantiff/PheKnowLator/blob/master/notebooks/Tutorials/entity_search/Entity_Search.ipynb
SPARQL Endpoint	http://sparql.pheknowlator.com/

^aGithub Notebook Directory:<https://github.com/callahantiff/PheKnowLator/blob/master/notebooks/>.

Acronyms: GCP (Google Cloud Platform); GCS (Google Cloud Storage); KG (knowledge graph); PKT (PheKnowlator); PKT-KG (PheKnowLator knowledge graph construction software).

Supplementary Table 4. PheKnowLator Ecosystem Evaluation Resources.

Resource	URL
Survey of Open Source KG Construction Software	
GitHub Scraper	https://gist.github.com/callahantiff/0ae1c00df9bec7228be3f6bda5466d73
Survey	http://dx.doi.org/10.5281/ZENODO.5790040
PKT Human Disease KGs	
PKT-KG Zenodo Release	https://zenodo.org/record/4685943#.YvLai-zMLOs
PyPi Release	https://pypi.org/project/pkt-kg/2.1.0/
Build Documentation	https://github.com/callahantiff/PheKnowLator/builds/README.md
Data Source Descriptions	https://github.com/callahantiff/PheKnowLator/wiki/v2-Data-Sources#data-sources
GCS Bucket ^a	https://console.cloud.google.com/storage/browser/pheknowlator/archived_builds/release_v2.1.0/build_01MAY2021
Zenodo Archive	https://zenodo.org/communities/pheknowlator-benchmark-human-disease-kg
GitHub Actions Workflows	https://github.com/callahantiff/PheKnowLator/.github/workflows/kg-build-part1.yml https://github.com/callahantiff/PheKnowLator/.github/workflows/kg-build-part2.yml
Docker	https://github.com/callahantiff/PheKnowLator/builds/Dockerfile.phases12 https://github.com/callahantiff/PheKnowLator/builds/Dockerfile.phase3
Data Sources	https://github.com/callahantiff/PheKnowLator/builds/data_to_download.txt
PKT-KG Input Dependencies ^a	https://zenodo.org/record/7026644/files/edge_source_list.txt https://zenodo.org/record/7026644/files/ontology_source_list.txt https://zenodo.org/record/7026644/files/resource_info.txt
Build Metadata ^a	https://zenodo.org/record/7026640/files/downloaded_build_metadata.txt https://zenodo.org/record/7029940/files/edge_source_metadata.txt.zip https://zenodo.org/record/7029940/files/ontology_source_metadata.txt.zip https://zenodo.org/record/7026644/files/preprocessed_build_metadata.txt
Ontology QC Report ^a	https://zenodo.org/record/7026644/files/ontology_cleaning_report.txt
Build Logs ^{a,b}	https://zenodo.org/record/7029940/files/pkt_builder_phases12_log.log.zip https://zenodo.org/record/7029940/files/pkt_build_log.log.zip
Data Files ^a	Original Data: https://zenodo.org/record/7026640 Processed Data: https://zenodo.org/record/7026644
KG Files ^a	<i>Class-based Knowledge Model</i> Standard Relations without Semantic Abstraction: https://zenodo.org/record/7029958 Standard Relations with Semantic Abstraction: https://zenodo.org/record/7029954 Inverse Relations without Semantic Abstraction: https://zenodo.org/record/7029942 Inverse Relations with Semantic Abstraction: https://zenodo.org/record/7029922 <i>Instance-based Knowledge Model</i> Standard Relations without Semantic Abstraction: https://zenodo.org/record/7029942 Standard Relations with Semantic Abstraction: https://zenodo.org/record/7029940 Inverse Relations without Semantic Abstraction: https://zenodo.org/record/7029946 Inverse Relations with Semantic Abstraction: https://zenodo.org/record/7029920

^aGCS data: https://console.cloud.google.com/storage/browser/pheknowlator/archived_builds/release_v2.1.0/build_01MAY2021.

^bBuild logs are found in each of the knowledge graph directories on GCS and the Zenodo Community for PKT Human Disease KG builds (<https://zenodo.org/communities/pheknowlator-benchmark-human-disease-kg>). The link provided is for the Instance-based Knowledge Model with Standard Relations and semantic Abstraction.

Acronyms: GCS (Google Cloud Storage); KG (knowledge graph); PKT-KG (PheKnowLator knowledge graph construction algorithm); QC (quality control).

Supplementary Table 5. Open-Source Knowledge Graph Construction Methods Survey Criteria.

Criteria	Description	Example Questions
Construction Functionality	An assessment of how well the method covers the steps needed to construct a knowledge graph from downloading and processing data and building edge lists to generating and outputting a KG	Is there functionality to download data? Can multiple types of KGs be constructed? Is preprocessing or filtering performed as part of the construction process?
Maturity	An assessment of the level, stage or development phase of a method	Is a versioning system in place? Have many releases been made? Are procedures in place to enable collaboration?
Availability	An assessment of the openness of a method and the ease of obtaining a copy of the method	Is the method licensed? What type of license is used?
Usability	An assessment of the efforts put in place to ensure that a user, with reasonable technical skills, could use the method	Is there a Wiki, Read the Docs, or GitPage associated with the method? Are there examples of how to use the method?
Reproducibility	An assessment of whether or not the method provides tools or resources to help reproduce the KG construction process and maintain the code base	What tools are provided to help enable reproducibility (e.g., Docker container, Jupyter Notebook, R Markdown)? Does the repository include any form of testing?

Note. The survey questions were adapted from <http://dx.doi.org/10.1109/aswec.2004.1290484>.

Supplementary Table 6. Open-Source Knowledge Graph Construction Methods.

Method	GitHub Repository ^a	Publication DOI	Primary Goal or Objective (from GitHub) ^b	Method Validation	Most Recent Repository Interaction ^c
Bio2BEL	bio2bel/	10.1101/631812v1	Bio2BEL uses the Biological Expression Language as a common schema for integrating a wide variety of biomedical databases including causal, correlative, and associative relationships between entities on the molecular, process, cellular, systems, and population levels	N/A	Within the last month
Bio2RDF	bio2rdf	10.1016/j.jbi.2008.03.004	Bio2RDF is an open-source project that uses Semantic Web technologies to build and provide the largest network of Linked Data for the Life Sciences	Examined impact of four transcription factors in Parkinson's disease	Within the last week
Bio4J	bio4j/bio4j	10.1101/016758	Bio4j aims to offer a platform for the integration of semantically rich biological data using typed graph models	Tool use demonstration; no formal biological validation	> 1 year
BioGrakn	graknlabs/biograkn	10.1007/978-3-319-61566-0_28	BioGrakn is based on GRAKN.AI, which is a deductive database in the form of a knowledge graph, allowing complex data modelling, verification, scaling, querying and analysis	Illustrative queries spanning precision medicine, text mining, and disease	Within the last month
Clinical Knowledge Graph (CKG)	MannLabs/CKG	10.1101/2020.05.09.084897	Clinical Knowledge Graph is a platform with twofold objectives: 1) build a graph database with experimental data and data imported from diverse biomedical databases and 2) automate knowledge discovery making use of all the information contained in the graph	Biomarker studies to demonstrate CKG use for clinical decision-making	Within the last month
COVID-19-Community	covid-19-net/covid-19-community	NA	The COVID-19-Community is a community effort to build a Neo4j knowledge graph that links heterogenous data about COVID-19	Tool use demonstration	Within the last week
Dipper	monarch-initiative/dipper	NA	Dipper is a Python package to generate RDF triples from common scientific resources	Tool use demonstration	Within the last week
Hetionet	hetio/hetionet	10.7554/eLife.26726	Hetionet is a hetnet — network with multiple node and edge (relationship) types — which encodes biology. Hetnet was designed for Project Rephetio	Predicted the probability of treatment for 209,168 compound–disease pairs	Within the last year
iASIS Open Data Graph	tasosnent/Biomedical-Knowledge-Integration	arXiv:1912.08633	iASIS is a framework to automatically retrieve and integrate disease-specific knowledge into an up-to-date semantic graph	Examined use with lung cancer, dementia, and Duchenne Muscular Dystrophy	Within the last 6 months

Method	GitHub Repository ^a	Publication DOI	Primary Goal or Objective (from GitHub) ^b	Method Validation	Most Recent Repository Interaction ^c
KG-COVID-19	Knowledge-Graph-Hub/kg-covid-19	NA	KG-COVID-19 is a flexible framework to ingest, integrate, and remix biomedical data to produce KGs for COVID-19 response. The framework can be applied to other problems in which siloed biomedical data must be quickly integrated for different biomedical research applications, including for future pandemics	Tool use demonstration	Within the last week
Knowledge Base Of Biomedicine (KaBOB)	UCDenver-ccp/kabob	10.1186/s12859-015-0559-3	KaBOB is a knowledge base of semantically integrated data. The system introduces five processes for semantic data integration including making explicit the differences between biomedical concepts and database records, aggregating sets of identifiers denoting the same biomedical concepts across data sources, and using declaratively represented forward-chaining rules to take information that is variably represented in source databases and integrating it into a consistent biomedical representation	Constructed a multi-species KG	Within the last year
Knowledge Graph Exchange (KGX)	NCATS-Tangerine/kgx	NA	KGX is a library and set of command line utilities for exchanging Knowledge Graphs that conform to or are aligned to the Biolink Model	Tool use demonstration	Within the last month
Knowledge Graph Toolkit (KGTK)	usc-isi-i2/kgtk/	arXiv:2006.00088	KGTK is a data science-centric toolkit to represent, create, transform, enhance and analyze KGs. KGTK represents graphs in tables and leverages popular libraries developed for data science applications, enabling a wide audience of developers to easily construct KG pipelines for their applications	Demonstrated functionality using Wikidata, DBpedia, and ConceptNet	Within the last week
ProNet	cran/ProNet	NA	ProNet provides functions for biological network construction, visualization and analyses, including topological statistics, functional module clustering, and GO-profiling	Examined H1N1 IAV-human protein-protein interactions	> 1 year
SEmantic Modeling machine (SEMi)	giuseppfutia/semi	10.1016/j.softx.2020.100516	SeMi (SEmantic Modeling machine) is a tool to semi-automatically build large-scale Knowledge Graphs from structured sources such as CSV, JSON, and XML files	Validated using advertising data	Within the last 6 months
PheKnowLator	PheKnowLator	10.1101/2020.04.30.071407	PheKnowLator (Phenotype Knowledge Translator) is a novel framework and fully automated Python 3 library explicitly designed for optimized construction of semantically-rich, large-scale biomedical KGs	Built and compared 12 benchmark KGs including construction performance	Within the last week

^aAll GitHub URLs begin with the following prefix: <https://github.com/>.

^bWhenever possible, descriptions of methods and tools were copied verbatim from the associated GitHub site, documentation, and/or manuscript.

^cThe most recent repository interaction was documented at the time of completing the survey, which was May 2020 (updated in June 2021).

Acronyms: KG (Knowledge Graph).

Supplementary Table 7. Open-Source Knowledge Graph Construction Survey - Functionality.

Method	Download Functionality	Edge list Functionality	Construction Functionality	Multiple KG Types	Other KG Construction Functionality	Process Ontology Data	Process Linked Open Data	Process Experimental Data	Process Clinical Data	Data Processing Limits
Bio2BEL	Yes	Yes	Yes	Yes	The entire PyBEL ecosystem tools are all available for all graphs generated by Bio2BEL	Yes	Yes	Yes	Yes	No
Bio2RDF	Yes	Yes	Yes	Yes	Talend RESTful API; community ontology mappings; SPARQL query repository	Yes	Yes	Yes	Yes	No
Bio4J	Yes	Yes	Yes	Yes	Titan, Anguillos API	Yes	Yes	No	No	No
BioGrakn	No	No	Yes	Yes	Provides different types of API clients (Java, Python, Node.js) and a Grakn Workbase	No	Yes	Yes	Yes	No
Clinical Knowledge Graph (CKG)	Yes	No	Yes	No	Data preparation (filtering, imputation, formatting); data analysis (dimensionality reduction, visualization, hypothesis testing)	Yes	Yes	Yes	No	No
COVID-19-Community	Yes	Yes	Yes	No	Neo4J Browser	No	Yes	No	Yes	No
Dipper	Yes	Yes	Yes	Yes	SciGraph RESTful API Build KGs with evidence and provenance	Yes	Yes	Yes	No	No
Hetionet	Yes	No	Yes	No	Neo4J Browser Creates permuted KGs	Yes	Yes	Yes	Yes	No
iASIS Open Data Graph	Yes	Yes	Yes	No	Biomedical Harvesters; MedKnow	Yes	Yes	No	Yes	No
KG-COVID-19	Yes	Yes	Yes	No	Leverages BioLink	Yes	Yes	No	No	No
Knowledge Base Of Biomedicine (KaBOB)	Yes	Yes	Yes	Yes	Blazegraph	Yes	Yes	No	No	No
Knowledge Graph Exchange (KGX)	Yes	Yes	Yes	Yes	KG verified to confirm to the Biolink model, summary statistics	Yes	Yes	Yes	No	No
Knowledge Graph Toolkit (KGTK)	Yes	Yes	Yes	Yes	Data cleaning module, processes other KGs, KG querying modules, summary statistics, node embeddings	No	Yes	No	No	No
ProNet	No	Yes	Yes	No	KG visualization; enables topological analyses	No	Yes	Yes	No	No
SEmantic Modeling machine (SEMi)	No	Yes	Yes	Yes	Semantic type detector; weighted graph generator; semantic model builder and refiner; link predictor	Yes	Yes	No	No	No
PheKnowLator	Yes	Yes	Yes	Yes	Data download and preprocessing tools; ontology quality control tools; export node metadata; property graphs; SPARQL Endpoint	Yes	Yes	Yes	No	No

Note. For scoring, 1 point was awarded for an answer of “Yes” and for the presence of other KG construction functionality.

Acronyms: KG (Knowledge Graph).

Supplementary Table 8. Open-Source Knowledge Graph Construction Survey - Availability.

Method	Open Source	License	Operating Systems	Coding Languages	External Dependencies
Bio2BEL	Yes	MIT	Linux, Windows, Mac OSX, Cloud-based systems and/or architectures	Python, SQL	Bioregistry, PyOBO, Bioversions, various other standard Python packages
Bio2RDF	Yes	MIT Apache 2.0 CC0-1.0	Linux, Windows, Mac OSX	Java, JavaScript, Shell, OWL	Virtuoso, GIT
Bio4J	No	AGPL-3.0	Linux, Windows, Mac OSX, Cloud-based systems and/or architectures	Java, Scala	Anguillos, AWS EC2/S3, Titan
BioGrakn	No	None	Linux, Windows, Mac OSX, Cloud-based architectures	Python, Java, Node.js	GraknLabs, Maven
Clinical Knowledge Graph (CKG)	Yes	MIT	Linux, Windows, Mac OSX	Python	Java SE Runtime, Neo4j, R, Python 3.6
COVID-19-Community	No	MIT	Linux, Windows, Mac OSX	Python, Shell	Neo4J, Anaconda
Dipper	No	BSD-3	Linux, Windows, Mac OSX	Python, TSQL	
Hetionet	Yes	CC0	Linux, Windows, Mac OSX	Python, Shell	Docker, Neo4J
iASiS Open Data Graph	No	Apache 2.0	Linux, Windows, Mac OSX	Python, Java	MongoDB, UMLS, ReVerb, MetaMap, SemRep, YAJL, Neo4J
KG-COVID-19	Yes	BSD-3	Linux, Windows, Mac OSX	Python	KGX, BioLink
Knowledge Base Of Biomedicine (KaBOB)	Yes	GPL	Linux, Windows, Mac OSX	Groovy, Clojure, Shell	Docker, Maven
Knowledge Graph Exchange (KGX)	Yes	BSD-3	Linux, Windows, Mac OSX	Python	Docker, BioLink
Knowledge Graph Toolkit (KGTK)	Yes	MIT	Linux, Windows, Mac OSX	Python	Anaconda, mlr
ProNet	Yes	GPL (>=2)	Linux, Windows, Mac OSX	R	BioGrid, GO
SEmantic Modeling machine (SEMi)	Yes	GPL	Linux, Windows, Mac OSX	Python, JavaScript, Shell	Anaconda, Node.js (11.15.0), Java, Maven, Elasticsearch
PheKnowLator	Yes	Apache 2.0	Linux, Windows, Mac OSX, Cloud-based systems and/or architectures	Python, Java, Shell	OWL Tools

Note. For scoring, 1 point was awarded for an answer of “Yes” and for the presence of a license.

Supplementary Table 9. Open-Source Knowledge Graph Construction Survey - Usability.

Method	README	Wiki, Docs, or GitPage	Example Use	Tutorials	Install Tools	Method Use Resources	Sample Data	Handles Different Sized Data	Output Types	Adoption Indicators
Bio2BEL	Yes	Yes	Yes	No	PyPI Maven	None	Yes	Yes	NetworkX, Cytoscape, text files, n-triples, Biological Expression Language, several IO formats	Yes
Bio2RDF	Yes	Yes	Yes	No	None	None	Yes	Yes	Virtuoso dump, OWL, nq	Yes
Bio4J	Yes	Yes	Yes	Yes	AWS S3	Anguillos API Titan	Yes	Yes	Titan	Yes
BioGrakn	Yes	Yes	Yes	Yes	None	Grakn Clients	Yes	Yes	Grakn KG output types	Yes
Clinical Knowledge Graph (CKG)	Yes	Yes	Yes	Yes	Docker	Jupyter Notebook Docker	Yes	Yes	Neo4j	Yes
COVID-19-Community	Yes	No	Yes	Yes	Jupyter Notebook	Jupyter Notebooks	Yes	Yes	Neo4J, CSV	Yes
Dipper	Yes	Yes	Yes	Yes	PyPI	Jupyter Notebooks	Yes	Yes	TTL, Neo4J, TSV	Yes
Hetionet	Yes	Yes	Yes	Yes	Jupyter Notebook	Jupyter Notebook Docker	Yes	Yes	JSON, Neo4J, TSV, and Matrix	Yes
iASIS Open Data Graph	Yes	Yes	Yes	No	None	None	No	Yes	JSON, CSV, Neo4J, MongoDB	Yes
KG-COVID-19	Yes	Yes	Yes	Yes	None	None	Yes	Yes	RDF, TSV	Yes
Knowledge Base Of Biomedicine (KaBOB)	Yes	Yes	Yes	Yes	Docker	Docker	Yes	Yes	RDF/XML	Yes
Knowledge Graph Exchange (KGX)	Yes	Yes	Yes	Yes	PyPI Docker	None	Yes	Yes	OWL or RDF/XML, NetworkX, text files, n-triples files, tar, csv, graphML, TTL, JSON, RQ, RSA	Yes
Knowledge Graph Toolkit (KGTK)	Yes	Yes	Yes	Yes	Jupyter Notebook Docker	Jupyter Notebook Docker	Yes	Yes	n-triples files, JSON, Neo4J, GML	Yes
ProNet	No	Yes	Yes	Yes	CRAN	R Markdown	Yes	Yes	R data frame object (rda)	No
SEmantic Modeling machine (SEMI)	Yes	Yes	Yes	No	PyPI	None	Yes	Yes	OWL or RDF/XML files, graph, json, TTL	No
PheKnowLator	Yes	Yes	Yes	Yes	PyPI Jupyter Notebook Docker	Jupyter Notebook Docker	Yes	Yes	RDF/XML, NetworkX, a text files, n-triples, JSON	Yes

Note. For scoring, 1 point was awarded for an answer of “Yes” and for the presence of tools to run and install the method.

Supplementary Table 10. Open-Source Knowledge Graph Construction Survey - Maturity.

Method	Multiple Releases	Release Count	Method Published	Collaboration Encouraged	Collaboration Procedures
Bio2BEL	Yes	1	Yes	Yes	Yes
Bio2RDF	Yes	2	Yes	Yes	No
Bio4J	Yes	100	Yes	No	Yes
BioGrakn	Yes	1	Yes	No	No
Clinical Knowledge Graph (CKG)	No	0	Yes	Yes	Yes
COVID-19-Community	No	0	No	Yes	Yes
Dipper	Yes	4	No	No	No
Hetionet	No	1	Yes	Yes	No
iASIS Open Data Graph	No	0	Yes	No	No
KG-COVID-19	No	0	No	Yes	Yes
Knowledge Base Of Biomedicine (KaBOB)	No	1	Yes	No	No
Knowledge Graph Exchange (KGX)	No	0	No	No	No
Knowledge Graph Toolkit (KGTK)	Yes	3	Yes	Yes	Yes
ProNet	Yes	1	Unclear	No	No
SEmantic Modeling machIne (SEMi)	No	0	Yes	No	No
PheKnowLator	Yes	1	Yes	Yes	Yes

Note. For scoring, 1 point was awarded for an answer of “Yes” and for the presence of at least one release.

Supplementary Table 11. Open-Source Knowledge Graph Construction Survey - Reproducibility.

Method	Reproducibility Tools	Install Services	Deployment Services	Maintainability Measures	Well-Documented Codebase	Actively Used Issue Tracker
Bio2BEL	CLI Tool	Yes	No	Yes	Yes	Yes
Bio2RDF	None	No	No	No	Yes	Yes
Bio4J	AWS S3 Titan distribution	No	Yes	No	Yes	Yes
BioGrakn	Grakn Tools	Yes	Yes	No	Yes	Yes
Clinical Knowledge Graph (CKG)	Jupyter Notebook Docker	No	No	No	Yes	Yes
COVID-19-Community	Jupyter Notebooks	No	No	No	Yes	Yes
Dipper	Jupyter Notebook	Yes	Yes	No	Yes	Yes
Hetionet	Jupyter Notebook Docker	No	No	No	Yes	Yes
iASIS Open Data Graph	None	Partial	No	No	Yes	Yes
KG-COVID-19	None	Yes	Yes	Yes	Yes	Yes
Knowledge Base Of Biomedicine (KaBOB)	Docker	Yes	Yes	No	Yes	Yes
Knowledge Graph Exchange (KGX)	Jupyter Notebook Docker	Yes	Yes	No	Yes	Yes
Knowledge Graph Toolkit (KGTK)	Jupyter Notebook Docker	Yes	Yes	Yes	Yes	Yes
ProNet	R Markdown	No	No	No	Yes	No
SEmantic Modeling machine (SEMi)	None	Yes	Yes	No	Yes	Yes
PheKnowLator	PyPI Docker Jupyter Notebook	Yes	Yes	Yes	Yes	Yes

Note. For scoring, 1 point was awarded for an answer of “Yes” and for the presence of at least one reproducibility tool.

Supplementary Table 12. PKT Human Disease Knowledge Graph Resources.

Resource	Provider	Filename
<i>OBO Foundry Ontologies</i>		
Chemical Entities of Biological Interest (ChEBI)	ChEBI	http://purl.obolibrary.org/obo/chebi.owl
Cell Ontology (CL) ^a	CL	http://purl.obolibrary.org/obo/uberon/ext.owl
Cell Line Ontology (CLO)	CLO	http://purl.obolibrary.org/obo/clo.owl
Gene Ontology (GO)	GO	http://purl.obolibrary.org/obo/go.owl
Human Phenotype Ontology (HPO)	HPO	http://purl.obolibrary.org/obo/hp.owl
Mondo Disease Ontology (Mondo)	Mondo	http://purl.obolibrary.org/obo/mondo.owl
Pathway Ontology (PW)	PW	http://purl.obolibrary.org/obo/pw.owl
Protein Ontology (PRO)	PRO	http://purl.obolibrary.org/obo/pr.owl
Relations Ontology (RO)	RO	http://purl.obolibrary.org/obo/ro.owl
Sequence Ontology (SO)	SO	http://purl.obolibrary.org/obo/so.owl
Uber-Anatomy Ontology (Uberon)	Uberon	http://purl.obolibrary.org/obo/uberon/ext.owl
Vaccine Ontology (VO)	VO	http://purl.obolibrary.org/obo/vo.owl
<i>Edge Data Sources</i>		
chemical-disease; chemical-phenotype	CTD	CTD_chemicals_diseases.tsv
chemical-gene; chemical-protein	CTD	CTD_chem_gene_ixns.tsv
chemical-biological process; chemical-cellular component; chemical-molecular function	CTD	CTD_chem_go_enriched.tsv
gene-pathway	CTD	CTD_genes_pathways.tsv
chemical-pathway	Reactome	ChEBI2Reactome_All_Levels.txt
biological processes-pathway; pathway-cellular component; pathway-molecular function	Reactome	gene_association.reactome
protein-pathway	Reactome	UniProt2Reactome_All_Levels.txt
disease-phenotype	HPO	phenotype.hpoa
gene-disease; gene-phenotype	DisGeNET	Curated_gene_disease_associations.tsv
gene-gene	Gene MANIA	COMBINED.DEFAULT_NETWORKS.BP_COMBINING.txt
gene-protein; gene-transcript; transcript-protein	PheKnowLator	^b Custom build files
protein-anatomy; protein-cell; transcript-anatomy; transcript-cell	HPA GTEEx	^b proteinatlas_search.tsv.gz GTEEx_Analysis_2017-06-05_v8_RNASeQCv1.1.9_gene_median_tpm.gct
protein-biological process; protein-cellular component; protein-molecular function	GO	goa_human.gaf
protein-pathway	Reactome	UniProt2Reactome_All_Levels.txt
protein-protein	STRING	9606.protein.links.v11.0.txt
variant-gene; variant-disease; variant-phenotype	ClinVar	variant_summary.txt
protein-catalyst; protein-cofactor	UniProt	uniprot-cofactor-catalyst.tab (^c query)

Resource	Provider	Filename
<i>Curation, Identifier Mapping, and Filtering Data</i>		
MeSH to ChEBI	MeSH CheBI	mesh2021.nt names.tsv
DOID, OMIM, Orphanet, ICD9, ICD10, UMLS, MeSH to Mondo and HP	DisGeNET HPO Mondo	disease_mappings.tsv database cross-references and synonyms
Ensembl, HGNC, Gene Symbol to Entrez STRING, UniProt to PR Gene Symbol, Entrez, HGNC to Ensembl	HGNC Ensembl Entrez PRO UniProt	hgnc_complete_set.txt Homo_sapiens.GRCh38.102.gtf.gz Homo_sapiens.GRCh38.102.uniprot.tsv.gz Homo_sapiens.GRCh38.102.entrez.tsv.gz Homo_sapiens.gene_info.gz promapping.txt uniprot_identifier_mapping.tab (^c query) ^b genomic_typing_dict.pkl
Tissues (string names) to Uberon Cell types (string names) to CL and CLO	PheKnowLator	^d zooma_tissue_cell_mapping_04JAN2020.xlsx
Reactome to PW	Reactome ComPath	ReactomePathways.txt gene_association.reactome ChEBI2Reactome_All_Levels.txt Compath_canonical_pathway_mappings.txt kegg_reactome.csv
genes, transcripts, and variants to SO	PheKnowLator	^b genomic_sequence_ontology_mappings.xlsx
Human PRO identifiers	PRO	human_pro_classes.html (^c query)

Note. Sources are reported for the v2.1.0 knowledge graphs (built May 2021). The full URLs are provided here:

https://github.com/callahantiff/PheKnowLator/blob/549e6e1e882e9ea579508ae24a90e64d962deb8c/builds/data_to_download.txt. The files for each source are provided in the PheKnowLator GCS Bucket (https://console.cloud.google.com/storage/browser/pheknowlator/archived_builds/release_v2.1.0/build_01MAY2021) and the PKT Human Disease KG Zenodo Community (v2.1.0_01MAY2021; <https://zenodo.org/communities/pheknowlator-benchmark-human-disease-kg>). The Wiki provides a description of each source and it's licensing: <https://github.com/callahantiff/PheKnowLator/wiki/v2-Data-Sources#data-sources>.

^aThe Cell Ontology is included with the extended version of Uberon.

^bCustom files built from processing the HGNC, ensembl, Entrez, PR, STRING, and UniProt data. See `data_preprocessing.py` in the `builds` directory or the `Data_Preparation.ipynb` Jupyter Notebook in the `Notebook` directory on GitHub for more details.

^cSee https://github.com/callahantiff/PheKnowLator/blob/549e6e1e882e9ea579508ae24a90e64d962deb8c/builds/data_to_download.txt to obtain the full queries.

^dManual annotation file built using Zooma (<https://www.ebi.ac.uk/spot/zooma/>).

Acronyms: CL (Cell ontology); CLO (Cell Line Ontology); ChEBI (Chemical Entities of Biological Interest); CTD (Comparative Toxicogenomics Database); DOID (Human Disease Ontology); GCS (Google Cloud Storage); GO (Gene Ontology); HGNC (Human Gene Nomenclature Committee); HPO (Human Phenotype Ontology); HPA (Human Protein Atlas); ICD (International Classification of Diseases); MeSH (Medical Subject Headings); Mondo (Mondo Disease Ontology); OMIM (Online Mendelian Inheritance in Man); PKT (PheKnowlator); PRO (Protein Ontology); PRO (Protein Ontology); PW (Pathway Ontology); SO (Sequence Ontology); VO (Vaccine Ontology); Uberon (Uber-Anatomy Ontology); UMLS (Unified Medical Language System).

Supplementary Table 13. Application of Data Quality Checks to OBO Foundry Ontologies.

Statistics ^a	CLO	ChEBI	GO	HPO	Mondo	PRO ^b	PW	RO	SO	Uberon	VO	Merged ^c
Pre-Processed Statistics												
Edges	1,387,096	5,264,571	1,425,434	884,999	2,313,343	2,079,356	35,291	7,970	44,655	752,291	86,454	13,746,883
Classes	111,712	156,098	62,237	38,843	55,478	148,243	2,642	116	2,910	28,738	7,089	548,947
Individuals	41	0	0	0	18	0	0	5	0	0	165	195
Object Properties	116	10	9	231	331	12	1	604	50	242	232	847
Annotation Properties	192	37	53	257	119	11	19	106	41	284	97	656
Connected Components	7	1	2	1	1	3	1	3	1	2	5	8
Data Quality Check Errors												
Value Errors	1	0	0	0	0	0	0	0	0	0	0	0
Identifier Errors	0	0	0	0	0	0	0	0	0	0	2	2
Deprecated Entities	2	18,506	6,430	304	2,305	0	42	11	341	1,570	0	0
Obsolete Entities	13	0	0	0	0	0	0	1	0	0	0	0
Punning	16	0	0	0	0	0	0	0	0	0	0	8
Consistency ^d	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	---
Semantic Heterogeneity	---	---	---	---	---	---	---	---	---	---	---	7
Identifier Alignment	---	---	---	---	---	---	---	---	---	---	---	23,624
Post-Processed Statistics												
Edges	1,422,153	5,190,485	1,343,218	885,379	2,277,425	2,079,356	34,901	7,873	41,980	734,768	89,764	13,748,009
Classes	111,696	137,592	55,807	38,530	52,937	148,243	2,600	115	2,569	27,170	7,085	545,259
Individuals	33	0	0	0	17	0	0	5	0	0	165	188
Object Properties	112	10	9	231	330	12	1	594	50	238	232	846
Annotation Properties	187	37	53	257	119	11	19	106	41	284	97	656
Connected Components	7	1	2	1	1	3	1	3	1	2	5	8

Note. The OBO Foundry ontologies reported above apply to the PKT Human Disease KG v2.1.0. The extended version of Uberon used in this graph imports the full version of the Cell Ontology.

^aThe numbers for the ontologies are calculated using the versions of the ontologies which include all imported ontologies referenced by the primary ontology. This means that the counts of classes include all Web Ontology classes used for logical definitions, not only those that are explicitly part of the primary ontology's namespace.

^bThe PRO version references the human (NCBITaxon_9606) subset created for the PheKnowLator ecosystem.

^cMerged represents all of the OBO Foundry ontologies merged into a single ontology.

^dConsistency was evaluated using the ELK reasoner. The reasoner was only applied to individual OBO Foundry ontologies.

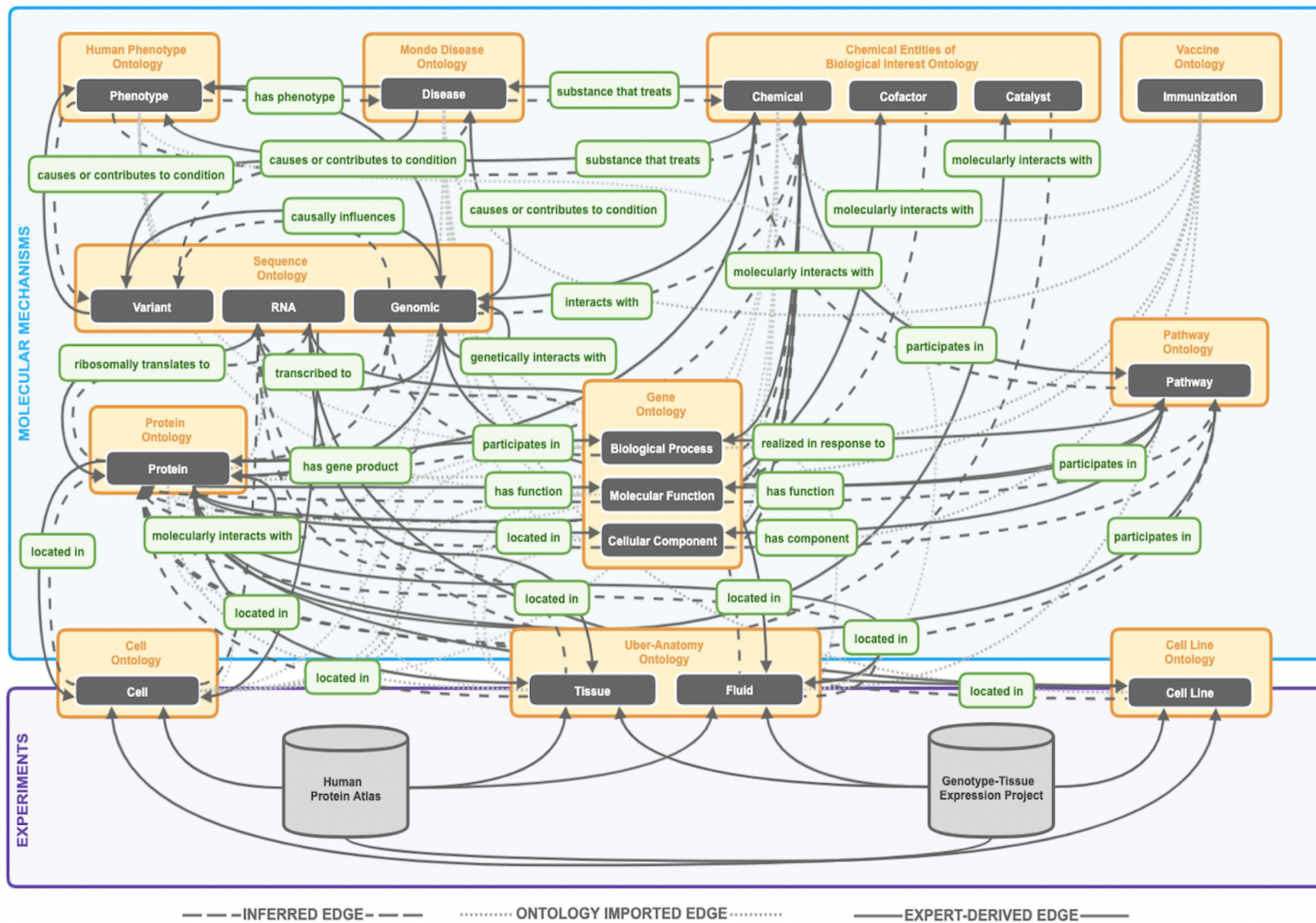
Acronyms: OBO (Open Biological and Biomedical Ontologies); CLO (Cell Line Ontology); ChEBI (Chemical Entities of Biological Interest); GO (Gene Ontology); HPO (Human Phenotype Ontology); Mondo (Mondo Disease Ontology); PRO (Protein Ontology); PW (Pathway Ontology); RO (Relation Ontology); SO (Sequence Ontology); Uberon (Uber-Anatomy Ontology); VO (Vaccine Ontology).

Supplementary Table 14. PheKnowLator Knowledge Modeling Approaches.

<p>Example: Add <<EDNRB, Causes, ABCD syndrome>> to an ontologically-grounded knowledge graph.</p> <p>Challenge: EDNRB is not currently represented in an ontology. ABCD syndrome is a class in the Human Phenotype Ontology, and is included in the knowledge graph.</p> <p>Solution: Gene is a class in the Sequence Ontology and can be used to add EDNRB to the knowledge graph using two different strategies.</p>	
Instance-based Knowledge Model (ABox)	Class-based Knowledge Model (TBox)
<p>EDNRB, rdfs:subClassOf, Gene EDNRB, rdf:type, owl:Class</p> <p>UUID1, rdf:type, EDNRB UUID1, rdf:type, owl:NamedIndividual</p> <p>UUID2, rdf:type, ABCD syndrome UUID2, rdf:type, owl:NamedIndividual</p> <p>UUID1, Causes, UUID2</p>	<p>EDNRB, rdfs:subClassOf, Gene EDNRB, rdf:type, owl:Class</p> <p>UUID1, rdfs:subClassOf, EDNRB UUID1, rdfs:subClassOf, UUID2 UUID2, rdf:type, owl:Restriction UUID2, owl:someValuesFrom, ABCD syndrome UUID2, owl:onProperty, Causes</p>

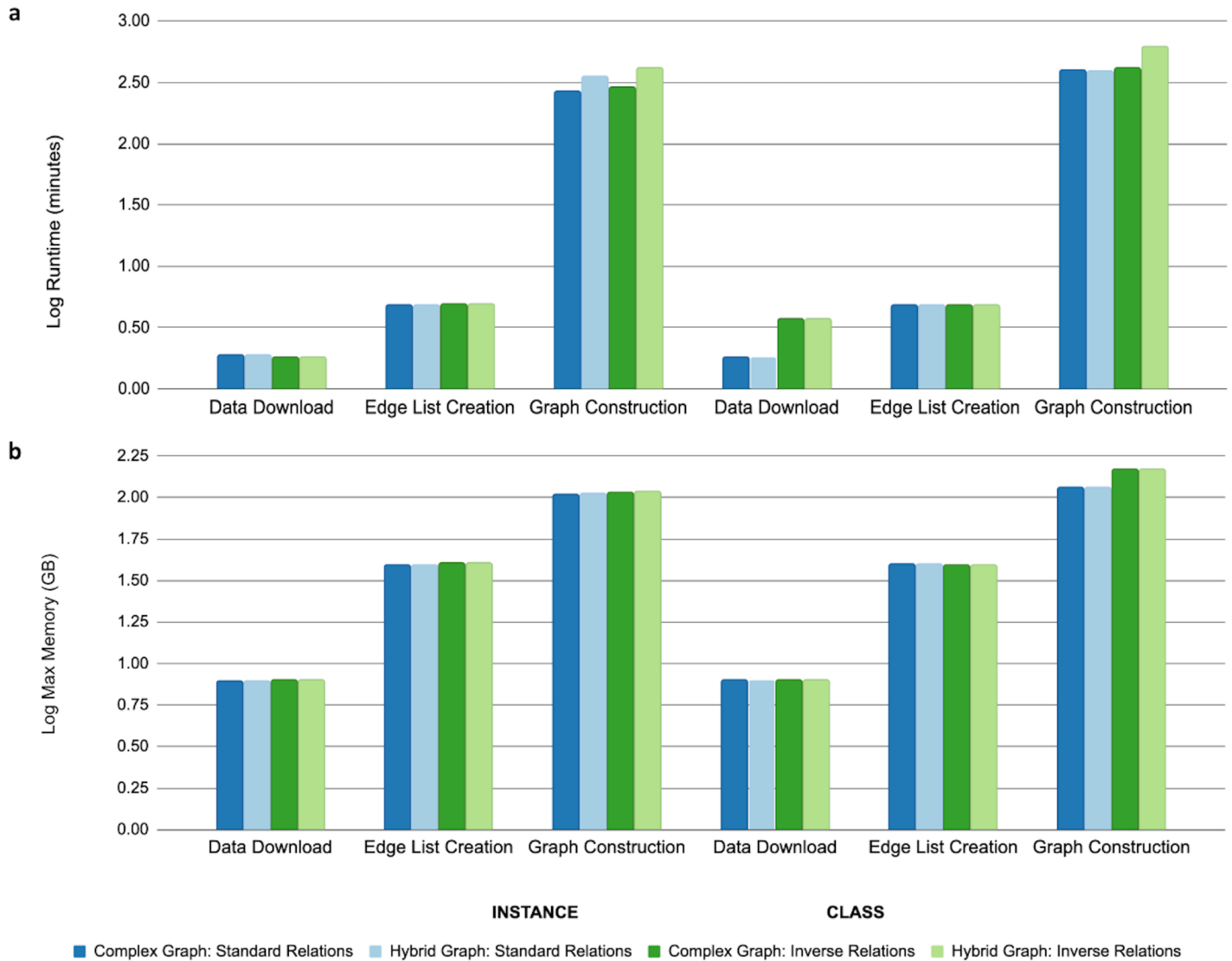
Note. UUID1 and UUID2 are blank nodes or existential variables.¹⁶² Pink highlighting is used for the EDNRB gene instance, yellow highlighting is used for the gene class, and green is used for the ABCD syndrome class.

Acronyms: EDNRB (endothelin receptor type B); OWL (Web Ontology Language); RDF (Resource Description Framework); RDFS (Resource Description Framework Syntax).







Supplementary Figure 1. Human Disease Mechanism Graph Knowledge Representation.

This figure illustrates the knowledge representation used to construct the human disease mechanisms knowledge graphs. The purple box represents experimental data and the blue box contains the molecular mechanisms created by integrating Open Biological and Biomedical (OBO) Foundry ontologies (gold and green). Edges between the ontologies are created by integrating other data sources that are not part of an OBO Foundry ontology (solid black lines). Dashed lines represent relationships that are inferred from the Relation Ontology and dotted lines represent relationships that exist between imported ontologies.



Supplementary Figure 2. PKT Human Disease Knowledge Graph Construction - Computational Performance.

This figure illustrates the (A) log runtime and (B) log max memory use (GB) performance for each build step with respect to the different build parameterizations or benchmarks provided by the PheKnowLator ecosystem. The ecosystem enables users to fully customize KGs generated by the Graph Construction build step through the following parameters: knowledge model (i.e., complex graphs constructed using class- or instance-based knowledge models), relation strategy (i.e., standard directed relations or inverse bidirectional relations), and semantic abstraction (i.e., transformation of complex graphs into hybrid graphs). The Data Download and Edge List Creation steps are the same regardless of how the Graph Construction step is parameterized. Computational performance was determined using an unreleased build (April 11, 2021) while testing the v.2.1 release.

 Findable	Unique Persistent Identifiers <ul style="list-style-type: none"> • Data: Original and processed data • Metadata: Logs and quality reports • Infrastructure: Compute and containers
 Accessible	Publicly Available <ul style="list-style-type: none"> • Storage: RESTful access to builds • Builds: Versioned on Docker Hub • Notebooks: User-friendly examples
 Interoperable	Standardized Resources <ul style="list-style-type: none"> • Data: Ontology alignment • Metadata: Provenance reporting • Output: Standard file formats
 Reusable	Detailed Documentation <ul style="list-style-type: none"> • Releases: Code, data, builds • Versioning: Semantic versioning • Licensing: Internal/external resources

Supplementary Figure 3. The PheKnowLator Ecosystem on FAIR Principles.

The PheKnowLator Ecosystem is built on the FAIR principles of Findability, Accessibility, Interoperability, and Reusability.¹³ Findability. Use of unique persistent identifiers for all downloaded and processed data, Docker containers, and compute instances and generation of metadata, reports, and logs. Accessibility. All resources are accessible via RESTful API access to a dedicated Google Cloud Storage Bucket, all builds are versioned, and Jupyter Notebooks are used to improve the usability of the Ecosystem resources. Interoperability. Built on Semantic Web standards, grounded in Open Biological and Biomedical Foundry ontologies, and adoption of standard identifiers for all resources. Reusability. Builds are automated, containerized, and deployed through GitHub Actions workflows, resources, scripts, and workflows are versioned using Semantic Versioning, the Ecosystem is licensed, and licensing constraints are enforced for all ingested data.