KG-BIOPORTAL: GRAPH-NATIVE ACCESS TO BIOMEDICAL ONTOLOGIES FOR NETWORK AND SYSTEMS BIOLOGY

J. Harry Caufield*, Nomi L. Harris, and Christopher J. Mungall

*Environmental Genomics and Systems Biology Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA *Email: jhc@lbl.gov*

Jennifer Vendetti, Michael V. Dorf, Alex Skrenchuk, and Mark A. Musen

Stanford Center for Biomedical Informatics Research, Stanford University, Stanford, CA 94305, USA

BioPortal is the world's most comprehensive repository of biomedical ontologies, hosting 1,549 ontologies with more than 15 million classes and over 100 million cross-ontology mappings. Network-based analysis in biology and medicine increasingly relies on knowledge graphs (KGs) that combine experimental observations with curated knowledge, but the ontological knowledge on which such graphs depend has been difficult to obtain in graph-native form: it is spread across hundreds of separately maintained artifacts, serialized in different languages, and rarely accompanied by consistent node and edge typing or version-level provenance. We describe recent work that makes BioPortal knowledge-graph-friendly.

KG-Bioportal consists of an open-source workflow, the software that runs it, and the derived data collection it publishes. It converts BioPortal ontologies into standard KGX-format node and edge lists, categorizes entities using the Biolink Model, records the source and submission version of every ontology, and publishes an independently downloadable graph for each one. Where an ontology imports others, a second graph with the import closure merged in is published beside it. The pipeline runs monthly on public infrastructure; the current collection covers 1,184 ontologies, 8,259,694 nodes, and 15,924,223 edges. For every ontology the software could not convert, it reports why.

Because graphs are released per ontology rather than as a monolith, users can compose made-for-purpose graphs containing only necessary domains. The median published graph has 412 nodes and 88% have fewer than 10,000 edges. We describe complementary improvements that support graph-based work: a machine-readable release index, and agent skills for retrieving and combining graphs, and a Model Context Protocol (MCP) server that lets AI agents query BioPortal without fabricating term identifiers. Finally we present a browser that indexes BioPortal-derived graphs alongside 161 community knowledge graphs, and our design for extending BioPortal to accept, search, and serve knowledge graphs directly.

Keywords: knowledge graphs; biomedical ontologies; data integration; semantic interoperability; BioPortal; Biolink Model; network analysis.

© 2026 The Authors. Open Access chapter published by World Scientific Publishing Company and distributed under the terms of the Creative Commons Attribution Non-Commercial (CC BY-NC) 4.0 License.

1. Introduction

A great deal of modern biomedical analysis is, in its roots, network analysis. Gene regulatory network inference, signaling pathway reconstruction, multi-omics integration, and causal modeling all share a conceptual lineage with more applied research tasks: defining a cohort from clinical notes, detecting a pharmacovigilance signal, harmonizing agricultural or environmental measurements, or indexing a literature corpus. In every case the analyst must be able to assert that an entity recorded in one place is the same entity as, or stands in a defined relationship to, an entity recorded in another. A transcript identifier, a drug trade name, or a phenotype extracted from a clinical note are referring expressions whose referents must be reconciled before any network built from them will mean anything. Ontologies (standardized, computable representations of the entities in a discipline) do this reconciling work[1]. They supply the identifiers, the definitions, the synonym sets, and the hierarchical and non-hierarchical relationships that let heterogeneous measurements be placed in a shared frame of reference.

Increasingly, that knowledge is consumed by software rather than read by people. Graph machine learning methods are now routine in biomedical discovery, and libraries such as GRAPE[2] operate on typed node and edge lists rather than on ontology documents, as do the AI-powered agentic systems now applied to annotation and curation. Both require the same thing: domain knowledge already in graph form, with consistent types and traceable provenance.

BioPortal has served as the bioinformatics community's principal repository of ontology knowledge for nearly two decades[3,4]. Launched by the National Center for Biomedical Ontology in 2005, BioPortal now houses 1,549 biomedical ontologies, 1,182 of them public, which together contain 15,293,440 classes and more than 101 million cross-ontology mappings, spanning molecular biology, clinical terminology, pharmacology, agriculture, environmental science, and the social and behavioral sciences. It is embedded in the routine operation of widely used research systems including REDCap[5] and the CEDAR Workbench[6]. BioPortal is deliberately open: any registered user may contribute an ontology. The repository includes all ontologies from the OBO Foundry along with a large selection of UMLS vocabularies and several hundred ontologies uploaded directly by their authors.

Despite this breadth, the knowledge BioPortal holds has historically been tricky to use in graph-based analysis. The obstacles are practical, and they are largely inherited from how ontologies are authored and published rather than introduced by the repository: BioPortal is faithful to what its contributors submit, and what they submit is not graphs. Ontologies arrive in several serializations, including OWL in any of its RDF encodings, OBO format, SKOS thesauri, and vocabularies derived from the UMLS, and a graph builder must handle each. For the OWL portion of the repository, much of the biologically interesting structure is carried in class axioms

rather than in simple triples, so a naive conversion loses exactly the non-hierarchical relationships that network analysis wants; the statement that a structure is part of some organ, for example, is written as an existential restriction inside a logical definition rather than as a direct edge. Node and edge types are inconsistent across sources, which frustrates any downstream method that expects a schema. Provenance is usually absent from the converted product, so a graph assembled six months ago is difficult to reproduce. The consequence is that most groups building biomedical KGs write and maintain their own ingest code for each ontology they need, an effort duplicated across the field and one that quietly encodes different conversion decisions into each graph.

2. KG-Bioportal: Rendering Ontologies as Knowledge Graphs

KG-Bioportal is three things under a single name. First, it is a *workflow*: a sequence of inventory, download, conversion, typing, and publication steps applied to every public BioPortal ontology. Second, it is the *software* that runs that workflow: a Python package with a command-line interface, executed on public infrastructure on a monthly schedule and runnable locally by anyone. Finally, it includes the *data collection* the workflow produces: one downloadable, independently versioned graph per ontology, published as release assets and indexed by a machine-readable manifest and a public browser. It is not a database, a query service, or a merge tool. Combining graphs is a separate operation that users perform with standard tooling (**Section 2.5**). KG-Bioportal's input is the BioPortal REST API. Its output, per ontology, is an archive containing node and edge tables. Each transformation is tracked in a publicly-accessible index.

2.1. *Design goals*

We designed KG-Bioportal around four principles, each responding to an issue described in the Introduction section.

Graph-native output. Every ontology is emitted as a pair of node and edge lists in the tab-delimited Knowledge Graph Exchange (KGX) format, a representation for property graphs already used across the KG-Hub ecosystem[7]. The output is directly loadable by common graph tooling without further parsing of OWL.

Modularity and composability. Rather than publishing a single merged KG, we publish one graph per ontology and leave composition to the user. Each ontology thus becomes a unit that can be selected, pinned to a version, cited, swapped out, and combined independently of the others.

Versioning and provenance. Each graph is identified by the BioPortal submission number (i.e., the identifier BioPortal uses to distinguish different uploads) of the ontology it came from, and every node and edge carries the source it was derived from.

Reproducibility and openness. The workflow is open source, runs unattended on a schedule on public infrastructure, and publishes both its products and the statistics describing them.

2.2. *The transformation pipeline*

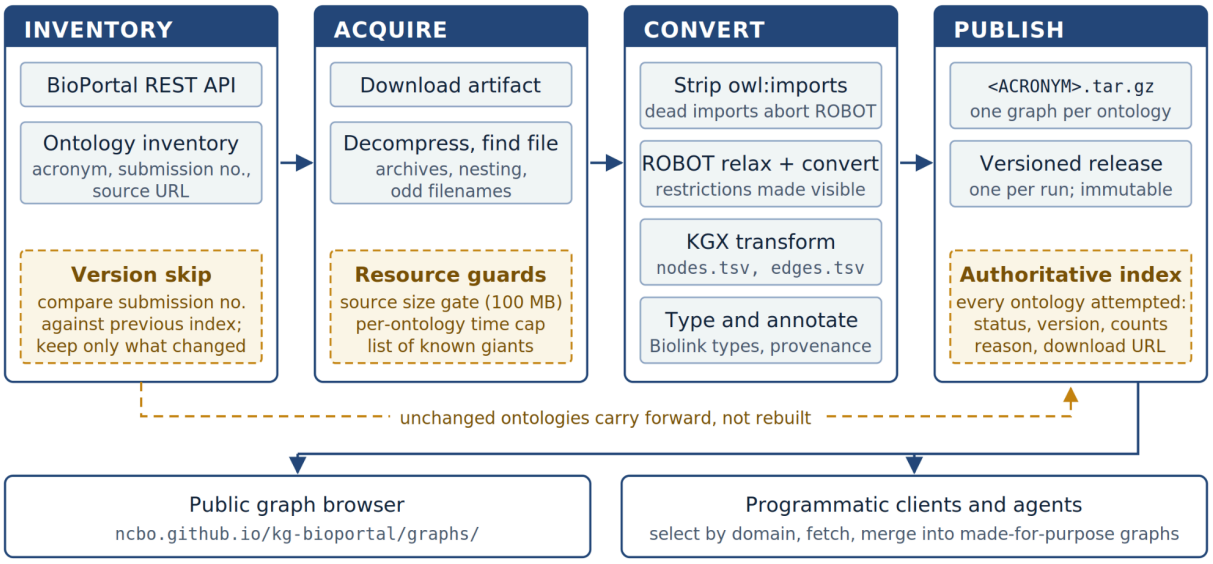


Fig. 1. The KG-Bioportal pipeline: inventory, acquisition, conversion, and publication, with the version-skip and resource-guard policies that bound each monthly run.

The pipeline workflow has four stages, illustrated in **Fig. 1** and described below. A scheduled run re-transforms only what changed and takes about ten minutes. A complete rebuild requires about two hours on twenty parallel runners.

The first stage retrieves the current inventory of ontologies via the BioPortal REST API, including each ontology's acronym, submission number, and download URL. Because each published graph records the submission it was built from, a scheduled run compares the new submission numbers against the previous index and re-transforms only the updated ontologies, carrying the rest forward untouched. This makes an unattended monthly build affordable.

The second stage downloads those ontologies. In practice this step carries most of the operational burden, because ontology files in an open repository are heterogeneous in ways that are difficult to anticipate: some arrive compressed, in several different archive formats; some archives contain a directory rather than a single file, or contain the ontology under an unexpected name; some submissions lack retrievable metadata altogether. The pipeline can detect and recover from these conditions so that one malformed ontology file will not halt the entire build process. Two resource guards apply here: a source-size gate (250 MB by default) and a per-ontology wall-clock cap, both configurable, which bound the cost of any single ontology on shared infrastructure.

The third stage performs the conversion. Each ontology is first processed with ROBOT[8], the standard command-line toolkit for ontology workflows. We apply ROBOT's *relax* operation, which does not add new relationships but makes existing ones visible to a converter that does no reasoning. Ontology authors commonly define a class with an equivalence axiom of the

genus-differentia form, stating that an X is a Y that is part of some Z; the existential restriction inside such a definition is invisible to a triple-level conversion. ‘Relax’ rewrites these into the weaker subclass form, asserting separately that X is a subclass of Y and that X is a subclass of the restriction *part of some Z*, and the subsequent conversion step renders the latter as an edge from X to Z labeled *part of*. Without this step a flattening conversion would yield little more than a subsumption tree, discarding the partonomic, regulatory, and participation relationships that encode most of the biology. The relaxed ontology is then transformed with KGX into node and edge tables, provenance fields recording the source ontology and submission are attached, and the resulting tables are written to a per-ontology archive.

Imports are handled in two ways, based on the OBO Foundry’s distinction between base and full ontology releases. Every ontology is published as a base graph such that the graph contains only what the ontology asserts and can be merged with other base graphs, avoiding duplicate nodes. When imports are declared and can be fetched, a full graph is published as well, with the import closure merged in. In the 2026-09-09 build, 263 of the 1,184 transformed ontologies declared imports, yielding 141 full graphs (120 failed due to unavailable imports; 2 of the full graphs exceeded the size cutoff). Nine ontologies contain only imports and are flagged accordingly in the KG-Bioportal index and pages.

KGX writes biolink:NamedThing on every node, because an OWL file says nothing about the Biolink Model[9]. The transform then assigns a more specific class wherever the graph's own evidence supports one, by four routes applied in order. A short table of seed terms lists classes whose Biolink meaning is not in doubt, such as GO's *biological_process* and MONDO's *disease*. Every class beneath a seed, along *subclass_of* or *skos:broader*, inherits its category. A category crosses an *exact_match* edge to a node that has none. Many ontologies hang off roots whose meaning is a fact about that ontology rather than about the term, such as an ICD chapter or an HGNC locus group. These roots were placed by reading each one and its subclasses in the published graph, one ontology at a time, with the assistance of an AI agent. Each placement is recorded in the repository with the reviewer, the date, the graph read, the subclass labels that served as evidence, and a confirmation field that stays empty until a maintainer has checked it. Roots that were read and refused are recorded with the reason. Nothing else is guessed: a node with no evidence stays *NamedThing*. Measured over the Sept. 9 build, the combined strategies assign a specific category to 52.2% of nodes. Every ontology's page in the browser reports the tally by route, with the disclosure that an AI agent assisted.

Products are published as release assets in the KG-Bioportal project repository. Hosting is organized as a set of releases plus one authoritative index: every run publishes its own release containing only what it built, and the index, republished in full each time, records for every ontology its status, name, version, submission number, node and edge counts, and a direct

download URL pointing at whichever release holds its artifact. The index is a small YAML file and is the intended entry point for programmatic users. Users who prefer to build graphs themselves (e.g., to pin a particular ontology version) can run the same pipeline locally against the BioPortal API.

2.3. *The resulting collection*

The collection is not one large graph stored in pieces, but a long tail of small, domain-specific graphs. As of the 2026-09-09 build, the collection comprises base graphs for 1,184 ontologies containing 8,259,694 nodes, representing ontology classes, and 15,924,223 edges, representing relationships among them. The size distribution matters: the median published graph has 412 nodes and 512 edges; 1,038 of the 1,184 graphs have fewer than 10,000 edges, and 1,139 have fewer than 100,000. The ten largest graphs hold 36% of all edges between them.

2.4. *What does not convert and why*

A build that quietly drops what it cannot handle may accomplish its purpose but would not support debugging. The index therefore carries an entry for every ontology the pipeline has attempted, successful or not, with a machine-readable reason for each failure, and the public browser (**Section 5**) gives every ontology graph a page whether or not an artifact exists behind it. **Table 1** provides details for the 2026-09-09 build. Of 1,303 ontologies attempted, 1,184 were converted. Of the 119 that were not, 31 were withheld by policy: 8 are available only under a license we do not hold, nearly all UMLS vocabularies, and are counted apart from failures because there is nothing to fix; 16 known giants, among them SNOMED CT, NCBI Taxonomy, RxNorm, MeSH, and NCIT, are skipped before download; and 7, including ChEBI, EFO, and FMA, exceed the 250 MB source-size gate. Of the 88 that failed, 78 failed on BioPortal's side, with no submission, no file, or no retrievable metadata, and 10 failed in conversion itself. The same transform code may be run locally, without the guards, on otherwise excluded ontologies.

Table 1. Outcome of the 2026-09-09 build for every ontology attempted.

Outcome	Reason recorded in the index	Ontologies
Converted	Base graph published (9 of these are import-only)	1,184
Withheld	License restricted (including UMLS vocabularies)	8
Withheld	Skipped before download due to file size limits	16
Withheld	Source file found to be over 250 MB limit	7
Failed	BioPortal side: no submission (27), no file (35), metadata not retrievable (16)	78
Failed	Conversion error: source unreadable by any parser (4), transform error (6)	10
Total attempted		1,303

The failure classes are not fixed. In the first public build, on 2026-08-02, conversion errors affected 189 ontologies, and the dominant cause was not malformed content but owl:imports. ROBOT resolves imports over the network when it loads an ontology, and a single dead, slow, or

access-restricted import URL aborts the entire conversion. Ontologies are often published with import targets that no longer resolve. The failure is silent for the author, whose local build environment has the imports cached. Stripping import declarations before invoking ROBOT rescued 121 of the 189 affected ontologies in that build, at the cost of leaving cross-ontology references as dangling edges that resolve when graphs are combined. Recording which stage lost each ontology, and what it said, then let the remaining errors be fixed one class at a time over the following month, and by the 2026-09-09 build transform errors were down to six. Import resolution proved an underappreciated obstacle to ontology reuse at scale: one change to one step moved roughly a tenth of the BioPortal inventory from unavailable to available. The cost of that change is measurable. In the 2026-09-09 base graphs, 5,265,221 of 15,924,223 edges (33%) touch a node the graph itself does not name: a term imported from another ontology, or an identifier the source refers to without defining. The share is 54% in the 263 graphs whose source declares imports and 28% in the rest, whose references were never imported to begin with. In the 141 ontologies for which a full graph was built, merging the import closure supplies a name for 36% of those edges' endpoints, which is the part attributable to stripping imports and the part the full graph repairs.

2.5. *Made-for-purpose graph assembly*

Publishing one graph per ontology is the feature that makes the collection more widely usable. We perform graph merges with the KGX cat-merge utility and document this as a recipe rather than implement it as a service. Node rows from the input graphs are unioned on the identifier column, so that two rows sharing a CURIE become a single node with their properties combined, and edge rows are concatenated and deduplicated on the subject-predicate-object triple. Nothing is collapsed on the basis of labels. Combining all graph-ready BioPortal ontologies this way is possible, and yields a giant merged KG of 8.3 million nodes and 15.9 million edges, but for most analyses it is the wrong object.

The difficulty a merged graph creates is not that it silently fuses distinct concepts, since identifier-keyed merging does not do that, but that it makes every sense of every term simultaneously present. For example, the label *nucleus* denotes the eukaryotic organelle in the Gene Ontology (`GO:0005634`), a neuroanatomical structure in the Zebrafish Anatomy and Development Ontology (`ZFA:0001665`), and the atomic nucleus in the Chemical Entities of Biological Interest (ChEBI) ontology (`CHEBI:33252`). All three terms survive the merge as separate nodes, and every operation that works from labels rather than from identifiers, including text annotation, lexical mapping, string search, and most embedding methods, must then arbitrate among them.

KG-Bioportal therefore supports selective assembly: a user selects the ontologies relevant to their specific question, retrieves those graphs by their download URLs from the index, and merges only those, by the procedure above. Graphs built this way typically contain fewer than 100,000 edges, two to three orders of magnitude smaller than the all-ontology graph. They are small enough to store in the local memory in most computing environments, to visualize, and to reason over interactively. Because the products conform to KGX, they can be easily combined with other KGX-format graphs such as those built under KG-Hub conventions[7] and those produced through the NCATS Translator project[10]. They can also be used as inputs for graph machine learning libraries such as GRAPE[2] for embedding and link prediction without an intervening translation step. KG-Bioportal intentionally restricts its scope to data, reserving analysis for existing graph libraries usable with this resource's common format. Selecting the ontologies that enter a graph is also a scoping decision recorded in the graph's construction rather than buried in an analysis script.

3. Supporting Infrastructure for Graph-based Bioinformatics

3.1. *Retrieving and composing graphs programmatically*

Systems biology workflows include steps that seem suitable for agentic approaches, such as assigning cell types to transcriptomic clusters, normalizing the columns of a sample sheet, or extracting entities from literature; however, generative AI has a well-documented tendency to fabricate plausible-looking ontology identifiers[11]. To avoid this, we developed *bioportal-mcp*, a server implementing the Model Context Protocol for AI agents that exposes BioPortal term search, property search, and usage analytics as tools an agent can call, so that the agent asks BioPortal what a term's identifier is instead of recalling or guessing it. This server is distributed through PyPI and can be added to an agent client with a single command. The server does not yet expose BioPortal's mapping service, though mappings may be obtained through the BioPortal REST API using the agent skills described below.

3.2. *Documenting procedures for reusable graph operations*

Documentation tells a reader what a resource contains but may not tell a piece of software what to do with it. An agent handed the KG-Bioportal index still has to rediscover which release holds an artifact and what columns the tables carry. Each detail is a place where a capable agent may produce confident nonsense that a human reviewer may not check.

We therefore distribute the procedural knowledge alongside the data, as three agent skills carried in the pipeline repository: *kg-bioportal-data* for locating and loading graphs, *kg-bioportal-merge* for combining them with cat-merge and reporting dangling cross-ontology edges, and *bioportal-for-kg-building* for using BioPortal's Annotator, Recommender, and mapping services as inputs. A fourth procedure belongs to the pipeline itself: a roots command that lays out

a graph's top-level classes with their subclass labels, which is the packet the category review of **Section 2.2** works from. Each carries helper scripts and data-model facts verified against the live collection, enabling agents to assemble made-for-purpose graphs. Agents are not compelled to use all data obtained from BioPortal (e.g., they are not required to use the provided mappings) but may instead work with graph data from multiple sources if specified.

4. Cross-domain Inference with Assembled Graphs

The value of rendering heterogeneous ontologies as composable graphs is best shown by example. Consider a clinician-researcher asking what adverse interactions might occur if a patient being treated for atrial fibrillation drinks grapefruit juice. A graph assembled from five ontologies answers the question along a single path. The Food Ontology supplies a term for raw grapefruit (`FOODON:03301702`). The Food Interactions with Drugs Evidence Ontology contributes a class describing an interaction between grapefruit and dronedarone (`FIDEO:000001707`). ChEBI provides the corresponding chemical entity (`CHEBI:50659`), conveniently annotated with the role "anti-arrhythmia drug," which both confirms the clinical context and makes the drug discoverable by mechanism rather than by name. A mapping links the ChEBI entry to its RxNorm equivalent (`RXNORM:233698`), where the drug acquires the trade names and dose forms that appear in prescribing records. The traversal from a food to a specific oral tablet crosses four independently maintained vocabularies; in the assembled graph it is a six-hop path, or longer if we extend the traversal to the Toxic Process Ontology (TXPO) (**Fig. 2**). The mapping that carries the path from ChEBI into RxNorm is an exact-label match, the kind of label-driven correspondence **Section 2.5** warns against. A lexical mapping like this is a candidate that a person or a rule accepts one at a time, as here, and not a basis on which a merge silently fuses nodes. Though neither ChEBI nor RxNorm graph transforms are in KG-Bioportal's hosted collection, their graphs may be built by running the same pipeline locally. This is the intended path for anything the hosted build does not carry.

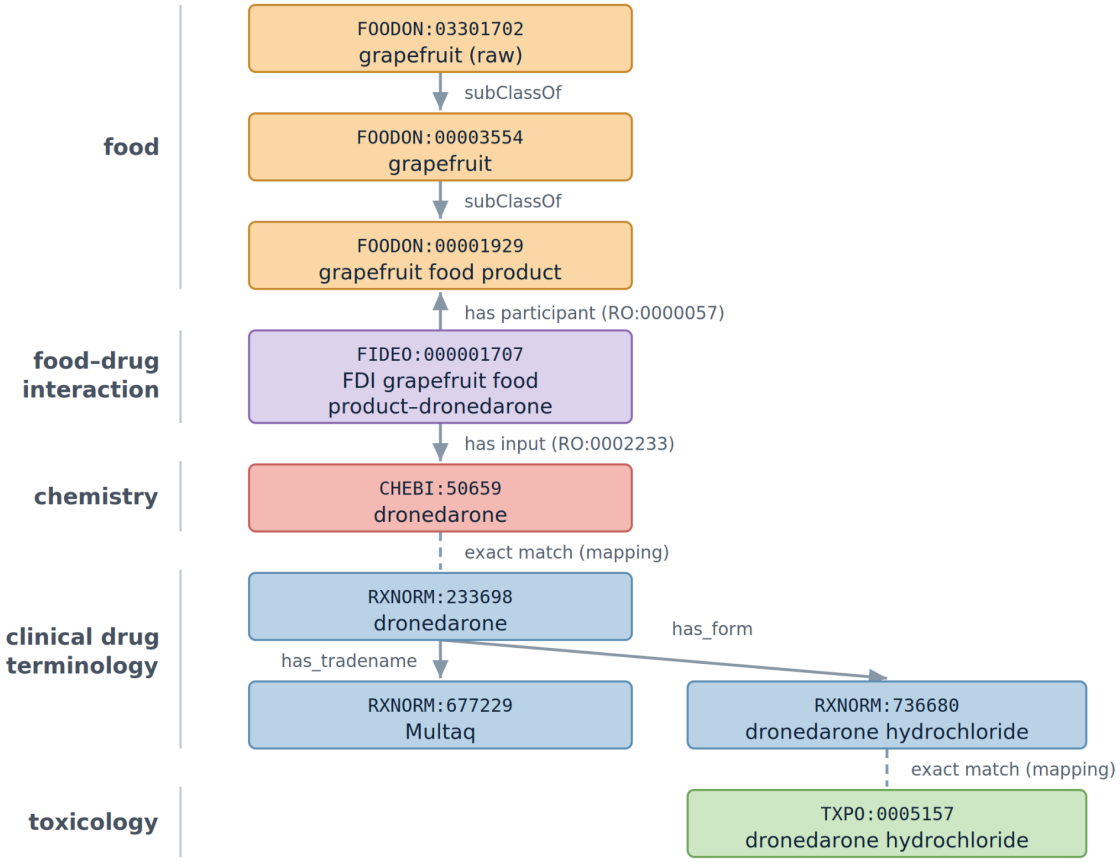


Fig. 2. A path through a made-for-purpose graph assembled from nine nodes from five BioPortal ontologies. Arrows between nodes represent typed relationships, along with their Relation Ontology (RO) identifiers where available. Dashed lines represent mappings obtained through BioPortal's LOOM method; all mappings shown here are exact label matches. Node color indicates the source ontology. The labels at left name the domain each source belongs to.

5. Toward a Repository for Knowledge Graphs

KG-Bioportal makes ontologies available as graphs. Biomedical KGs such as Hetionet[12], SPOKE[13], the Monarch Initiative KG[14], and PrimeKG[15] are central to work on drug repurposing, precision medicine, and genotype–phenotype inference, but they remain siloed, without a common entry point. The repository pattern that BioPortal established for ontologies has no counterpart for graphs, and the community has asked us to provide one.

BioPortal is built on RDF and stores its content in a triplestore, which gives us a graph database to build on but does not by itself solve the problem, because the biomedical KG community has largely moved toward labeled property graphs (LPGs). LPGs permit data to be attached to edges as well as nodes, so supporting them requires extending the BioPortal object model to associate information with edges. Once that is done, an edge asserting that imatinib

inhibits BCR-ABL1 can carry its interaction type, its supporting evidence, its source database, and its version, making the relationship not merely findable but interpretable.

We have deployed a first version of that entry point. It presents, in the familiar BioPortal browse idiom, a single filterable index over two populations that have until now been described in separate places: the BioPortal ontology entries produced by the pipeline and 161 community knowledge graphs, each with its domain, license, publications, and reported metrics, listed and faceted together. Every entry has a resource page, including the ontologies that failed to convert.

Storing graphs in the same backend as ontologies means that much existing BioPortal functionality extends to them at once. Nodes from submitted KGs become searchable exactly as ontology classes are, and may be returned by services such as the Annotator, which broadens the range of entities that BioPortal services can recognize. A user searching for quercetin would find the DrugCentral node (`drugcentral:3514`) even though no current BioPortal ontology contains it, and could follow its contraindication edges to Mondo kidney disease terms and its targets edges to Protein Ontology terms, with provenance carried on each edge. No comparable capability yet exists for KGs elsewhere.

6. Related Work

Several services besides BioPortal serve biomedical ontologies. EMBL-EBI's Ontology Lookup Service[16], the OBO Foundry[1], and Ontobee[17] all provide term-level search, resolution, and in some cases reasoning over ontologies. Each is designed for term retrieval rather than graph operations. Several projects have focused on rendering sets of OBO ontologies as graphs: Ubergraph materializes relaxed and reasoned OBO Foundry ontologies as one RDF graph[18]; the OBO Graphs JSON serialization[19] and semantic-sql[20] both give individual ontologies more graph-like or relational forms; and our own KG-OBO project served as a precursor to KG-Bioportal, as it performed OBO to KGX transformations across the full Foundry[21]. KG-Bioportal differs from each of these in scope and packaging. It covers nearly all of BioPortal rather than the OBO Foundry alone, involving a collection several times larger and containing a wider set of topics. KG-Bioportal also tracks the source and cause of all transformation errors.

KG-Bioportal is part of a broader ecosystem of tools and resources for working with biological and biomedical graph data. KNIT, for example, approaches reuse from the other direction, allowing users to explore an existing knowledge graph to decide which ontologies to reuse[22]. BioPortal's own Recommender tool serves the same purpose for free text. The KG-Bioportal graphs can then be assembled based on what the tool highlights. The knowledge graphs named in Section 5 also complement our work well.

Parts of the BioPortal infrastructure described in Sections 1 and 5 were reported in the 2025 BioPortal update. An early form of KG-Bioportal was presented at the 2025 Bioinformatics Open Source Conference. This paper is the first full description of the KG-Bioportal pipeline.

7. Discussion

The work described here rests on a simple observation: the biomedical community has invested enormous effort in building ontologies, and a second, largely separate effort in building knowledge graphs. The connection between the two has been left to individual groups to reconstruct repeatedly. Converting BioPortal's holdings into a consistent, typed, versioned graph collection turns a body of curated knowledge that was accessible but awkward into one that is directly usable by network methods. Running that conversion on a schedule, in public, with its failures on the record, yields a standing measurement of the biomedical ontology landscape’s graph-readiness.

Several limitations bound what we have achieved. Biolink categorization is only as good as the evidence in the source: 52.2% of nodes carry a category more specific than NamedThing, and the rest stay untyped rather than guessed. The ROBOT relax step and the conversion keep most but not all axiomatic structure. Measuring this on six OBO ontologies (GO, CL, HP, PATO, ENVO, and FOODON) by running the pipeline's own transform on their current sources and classifying every class axiom with a named subject (268,364 in all), we found that 99.0% of those axioms are a named superclass, a simple existential restriction, or an intersection of such. 99.97% of these are emitted as an edge. The remaining 1.0% (2,576 axioms, and 8.2% in ENVO, the most heavily axiomatized of the six) are nested or non-existential expressions: unions, complements, cardinality and value restrictions. KGX does not drop these outright but renders each as an edge to an unlabeled anonymous node. 1.3% of all edges in the collection are of that kind so the axiom is lost. Stripping owl:imports makes many more ontologies convertible but leaves cross-ontology references dangling until graphs are combined (**Section 2.4**), which is the right trade for a modular collection, and the full graphs exist for users who want a self-contained one.

Coverage is incomplete, for the reasons given in **Section 2.4**. The graphs inherit their sources' errors as faithfully as their content. A merged graph does not audit the ontologies that enter it, and a densely connected region is not a well-evidenced one. The provenance on every node and edge is helpful here, as a suspect assertion can be traced to the ontology and submission that made it. Finally, we have not yet shown that a graph composed from this collection serves a downstream task, such as link prediction, better than one built by a conventional OWL ingest. KG-Bioportal makes that comparison easy to set up and more reproducible. Use case comparisons and development of a full task-level benchmark will be performed in future work.

Some functionality will require additional consideration. A knowledge graph is not hierarchical and requires interface components able to display non-hierarchical structure. Services

such as the Recommender, which ranks ontologies by their fit to a body of text, need to be reconceived for graphs. Content overlap among graphs is expected and is not a problem to be resolved by merging: each graph can occupy its own space in the repository, retaining its own provenance and its own curatorial decisions, while shared entities become discoverable across graphs through BioPortal's mapping services. A user who finds a term in one graph should be able to see how other graphs represent it based on the evidence and context specific to each.

The trajectory we have set out points toward a repository in which ontologies and knowledge graphs are managed as two aspects of a single problem. Both define relationships among concepts; both are modular, versioned, community-maintained artifacts; both are consumed increasingly by software rather than by people. Treating them together lowers the barrier to entry for researchers who need reusable domain knowledge and, we hope, makes the graphs on which network-based biology increasingly depends as accessible and interoperable as the ontologies that underlie them.

BioPortal is freely available at https://bioportal.bioontology.org. KG-Bioportal is open source at https://github.com/ncbo/kg-bioportal, with prebuilt graphs and the collection index distributed through the repository's releases and browsable at https://ncbo.github.io/kg-bioportal/graphs/. The *bioportal-mcp* code is available at https://github.com/ncbo/bioportal-mcp and from PyPI. The software is released under a BSD license. Each graph is a derivative of one ontology and is redistributed under the ontology's own terms as recorded by BioPortal at its submission. Ontologies with licenses forbidding redistribution are not transformed (**Section 2.4**). Licenses for each transformed ontology are specified on KG-Bioportal's site and index except where no license has been specified.

Acknowledgments

This work has been supported in part by the National Institutes of Health [U24 GM143402, U54 HG004028, 5U01HG009453-03 (past support), 3OT2DB000009]; and by the Director, Office of Science, Office of Basic Energy Sciences of the US Department of Energy [Contract No. DE-AC0205CH11231 to JHC, NLH, and CJM]. We are grateful to Franz, Inc. (https://franz.com) for their generosity and technical support in enabling the integration of AllegroGraph with BioPortal, and to the BioPortal user and developer community, including the members of the OntoPortal Alliance, for their sustained contributions.

References

1. Jackson, R. *et al.* OBO Foundry in 2021: operationalizing open data principles to evaluate ontologies. *Database (Oxford)* **2021**, (2021).
2. Cappelletti, L. *et al.* GraPE: fast and scalable Graph Processing and Embedding. *Research Square* (2022) doi:10.21203/rs.3.rs-967145/v1.
3. Whetzel, P. L. *et al.* BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. *Nucleic Acids Res.* **39**, W541-5 (2011).
4. Vendetti, J. *et al.* BioPortal: an open community resource for sharing, searching, and utilizing biomedical ontologies. *Nucleic Acids Res.* **53**, W84–W94 (2025).
5. Harris, P. A. *et al.* The REDCap consortium: Building an international community of software platform partners. *J. Biomed. Inform.* **95**, 103208 (2019).
6. Musen, M. A. *et al.* The center for expanded data annotation and retrieval. *J. Am. Med. Inform. Assoc.* **22**, 1148–1152 (2015).
7. Caufield, J. H. *et al.* KG-Hub-building and exchanging biological knowledge graphs. *Bioinformatics* **39**, btad418 (2023).
8. Jackson, R. C. *et al.* ROBOT: A tool for automating ontology workflows. *BMC Bioinformatics* **20**, 407 (2019).
9. 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).
10. Fecho, K. *et al.* Announcing the biomedical data Translator: Initial public release. *Clin. Transl. Sci.* **18**, e70284 (2025).
11. Groza, T. *et al.* An evaluation of GPT models for phenotype concept recognition. *BMC Med. Inform. Decis. Mak.* **24**, 30 (2024).
12. Himmelstein, D. S. *et al.* Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *Elife* **6**, e26726 (2017).
13. Morris, J. H. *et al.* The scalable precision medicine open knowledge engine (SPOKE): a massive knowledge graph of biomedical information. *Bioinformatics* **39**, (2023).
14. Putman, T. E. *et al.* The Monarch Initiative in 2024: an analytic platform integrating phenotypes, genes and diseases across species. *Nucleic acids research* **52**, D938–D949 (2024).
15. Chandak, P., Huang, K. & Zitnik, M. Building a knowledge graph to enable precision medicine. *Sci. Data* **10**, 67 (2023).
16. McLaughlin, J., Lagrimas, J., Iqbal, H., Parkinson, H. & Harmse, H. OLS4: a new Ontology Lookup Service for a growing interdisciplinary knowledge ecosystem. *Bioinformatics* **41**, btaf279 (2025).
17. Ong, E. *et al.* Ontobee: A linked ontology data server to support ontology term dereferencing, linkage, query and integration. *Nucleic Acids Res.* **45**, D347–D352 (2017).
18. Balhoff, J. *et al.* Ubergraph: Integrating OBO Ontologies into a Unified Semantic Graph. *ICBO* 1–9 (2022).
19. Jacobsen, J. *et al. Geneontology/Obographs: Immutables and Splitting out OWL API*. (Zenodo, 2021). doi:10.5281/ZENODO.5070191.
20. Mungall, C. *Semantic-Sql: SQL and SQLite Builds of OWL Ontologies*. (Github, 2026).
21. Caufield, J. H., Reese, J. & Mungall, C. KG-OBO: Open bio-ontologies in knowledge graph form. (2022) doi:10.7490/f1000research.1119049.1.
22. Rodríguez-Revello, J., Barba-González, C., Rybinski, M. & Navas-Delgado, I. KNIT: Ontology reusability through knowledge graph exploration. *Expert Syst. Appl.* **228**, 120239 (2023).