

ROBOKOP KG and KGB: Integrated Knowledge Graphs from Federated Sources

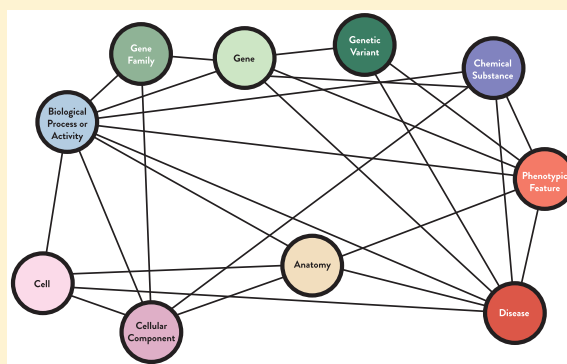
Chris Bizon,^{*,†} Steven Cox,[†] James Balhoff,[†] Yaphet Kebede,[†] Patrick Wang,[‡] Kenneth Morton,[‡] Karamarie Fecho,^{†,§} and Alexander Tropsha[§]

[†]Renaissance Computing Institute, University of North Carolina at Chapel Hill, Chapel Hill, North Carolina 27517, United States

[‡]CoVar Applied Technologies, Durham, North Carolina 27701, United States

[§]School of Pharmacy, University of North Carolina at Chapel Hill, Chapel Hill, North Carolina 27599, United States

ABSTRACT: A proliferation of data sources has led to the notional existence of an implicit Knowledge Graph (KG) that contains vast amounts of biological knowledge contributed by distributed Application Programming Interfaces (APIs). However, challenges arise when integrating data across multiple APIs due to incompatible semantic types, identifier schemes, and data formats. We present ROBOKOP KG (<http://robokopkg.renci.org>), which is a KG that was initially built to support the open biomedical question-answering application, ROBOKOP (Reasoning Over Biomedical Objects linked in Knowledge-Oriented Pathways) (<http://robokop.renci.org>). Additionally, we present the ROBOKOP Knowledge Graph Builder (KGB), which constructs the KG and provides an extensible framework to handle graph query over and integration of federated data sources.



1. INTRODUCTION

Data sources increasingly expose information via Application Programming Interfaces (APIs) that allow for structured query of biomedical knowledge. APIs may be based on experimental or curated data, collected in a variety of formats, and used to answer questions of interest to biomedical investigators, such as *what genes are involved with type 2 diabetes mellitus? what cell types are located in cardiac tissue? what diseases are associated with exposure to benzene?* Answers to these queries define a relationship between the input entity and the output entity. If this relationship is considered to be an edge in a Knowledge Graph (KG),^{1,2} then a distributed set of such APIs constitutes an implicit, federated KG, which then can be queried.³

Large, arbitrarily complex graph queries, ones that combine calls to multiple APIs and use results from one API as inputs to another, can be performed in principle, but any attempt to do so must deal with several challenges, including inconsistencies in data formats, identifier schemes, and semantic types.

Here, we present ROBOKOP Knowledge Graph Builder (KGB) and ROBOKOP KG, which are, respectively, software code that addresses these challenges and the database that is created by this code. These tools were initially created to support the open question-answering application ROBOKOP (Reasoning Over Biologically Oriented Knowledge Outcome Pathways),⁴ which is hosted at <http://robokop.renci.org> and supported by the National Center for Advancing Translational Sciences, as part of the Biomedical Data Translator (“Translator”) program.⁵ However, both tools are independently useful. ROBOKOP KGB builds KGs in response to user-

specified graph queries by parsing the queries into a series of type-driven API calls, between which the identifiers used in different systems are reconciled with one another in a process called “synonymization”. A general-purpose open biomedical ROBOKOP KG has been created by crawling over a large number of simple questions and combining the results into the single graph hosted at <http://robokopkg.renci.org>.

2. IMPLEMENTATION

2.1. BioLink Data Model and Type Hierarchy.

ROBOKOP KG is based on a set of semantic types, as defined in the BioLink data model.⁶ (A simplified version of the ROBOKOP KG database schema is provided as a graphical abstract, with nodes representing entities in the BioLink data model, and edges representing predicates or relationships between connected entities.) This model defines the high-level concepts between which relationships can be made, as well as a series of properties belonging to these concepts. The BioLink data model is hierarchical, with more specific concepts deriving from more general ones; for instance “cellular component” is derived from “anatomical entity”. The model also contains union terms that provide capabilities to, for example, group diseases and phenotypes. Entities are identified with conceptual terms from biomedical ontologies such as Gene Ontology (GO).⁷ ROBOKOP KG requires all nodes to be an instance of one or more BioLink data model types.

Received: August 21, 2019

Published: November 26, 2019

2.2. APIs and Clients. Individual knowledge sources are defined by APIs that take as input an entity (e.g., a disease) and return information about related entities (e.g., genes), as well as the nature of the relationship between them. Many formats of such APIs exist, including smartAPI⁸ and SPARQL or SQL endpoints. ROBOKOP KGB does not require a particular style or format for the APIs that it queries in order to allow for maximum generality in the queryable data sources. However, this approach requires that a client must be coded for each knowledge source. Data for which only flat files or RDF data can be acquired may be exposed using tools such as SmartBag⁹ or Data2Services.¹⁰ Because results are returned using types represented in the BioLink data model, we are able to create and apply mappings from the types returned from individual APIs to those contained in the BioLink data model.

The ROBOKOP KGB paradigm differs from earlier graph builders^{11–13} in its online approach. Previous approaches rely on downloading bulk data sets, which are curated and integrated together in bulk. In contrast, ROBOKOP KGB can respond to queries in which the data is fully federated, as long as an API exists to access it.

2.3. Concept Map. To calculate the services that must be called in order to produce answers, individual clients are annotated with their input and output types, thus forming a graph in which nodes represent types from the BioLink data model and edges represent the APIs that provide relationships between entities of those types. In the creation of this concept map, the BioLink data model allows inference of transitions involving entity superclasses and subclasses. For instance, if an API is annotated as returning entities of type “biological_process”, then the model implies that output entities are also members of the superclass “biological_process_or_activity”. Inferences can also proceed in reverse to identify subclasses.

2.4. Query Specification. Queries are specified using the Translator KG API standard specification (<https://github.com/NCATS-Tangerine/NCATS-ReasonerStdAPI/>), which defines a JSON-based representation of the requested graph. A node in the question graph represents an entity type; edges represent a desired relationship between entities of those types. Responses to the query will be in the form of a graph of instances that match the query in topology and entity types. Any node in the query graph may be bound to an identifier, so that any answer will contain only the entity specified at the corresponding location in the answer graph. Edge types can be specified to limit the allowed relationships between entities. Alternatively, if multiple edge types are specified, then edges that match any of the specified types will be returned.

2.5. Identifiers and Synonymization. APIs are generated by independent data providers and use different identifier systems for the entities they describe. The integration of APIs thus requires a reconciliation of different identifiers that represent the same entity. ROBOKOP KGB creates and maintains a set of equivalent identifiers and stores them in a Redis database for rapid retrieval during integration. Equivalent identifier sets are created per entity type. Where possible, we make use of pre-existing equivalences. For example, genes are synonymized using mappings from HGNC,¹⁴ and diseases are synonymized using equivalentTo and skos#exactMatch relationships from the Monarch Disease Ontology.¹⁵ Chemicals are iteratively identified starting with the use of InchiKey, as found in UniChem,¹⁶ then moving to SMILES for chemicals that are not expressible with InchiKey. For chemical substances lacking any structure, database cross references are used

(including the registry relationship in the case of MeSH¹⁷). Other types are synonymized using database cross-reference tags. Because such tags are inconsistently applied in different ontologies, specialized rules are crafted to prevent over eager identification, using pairs of vocabularies such as Human Phenotype Ontology and International Classification of Diseases.¹⁸ Given a set of equivalent identifiers, one is chosen as the node identifier for the output graph, using the vocabulary defined in the BioLink data model.⁶

2.6. Traversal. Query graphs are mapped to the concept transition model by matching query edges into model transitions and, thereby, into data source invocations. The traversal collects the minimal subgraph of federated data that contains every answer to the query. The transition model allows multiple paths per transition; it differs from a strict nondeterministic finite automaton (NFA) in that start and end states are not explicitly modeled, and transitions entailing ontological relationships are implicit. An accepted traversal of the NFA for a question compiles a stack of service invocations, the execution of which invokes services that produce triple instances that ontologically conform to the query pattern. Queries that are not traversals of the NFA are invalid with respect to the grammar and are rejected. ROBOKOP KGB manages results across service calls by resolving identifier synonyms, managing equivalent identifiers in subsequent calls, and caching results in a Redis key-value store in order to improve performance.

2.7. Integrated Database. The output from a query is written to a Neo4j database that integrates the data returned from all previous queries. Running multiple queries allows both the Neo4j and Redis caches to grow through query answers. Alternatively, a set of queries such as “return every gene-disease relationship” may be asked systematically, thereby effectively crawling all of the data sources, integrating the data, and storing the data in the Neo4j and Redis databases. The output of such a crawling campaign is ROBOKOP KG. ROBOKOP KG consists of data drawn either directly or indirectly from numerous data sources, including DrugBank,¹⁹ DrugCentral,²⁰ Aeolus,²¹ Comparative Toxicogenomics Database,²² PubChem,²³ Panther,²⁴ UniChem,¹⁶ ChEMBL,²⁵ Chemical Entities of Biological Interest,²⁶ mychem.info,²⁷ Monarch,²⁸ Monarch Disease Ontology,¹⁵ Human Phenotype Ontology,²⁹ GO,³⁰ QuickGO,³¹ AmiGO,³² Pharos,³³ ClinGen,³⁴ ClinVar,³⁵ GWAS Catalog,³⁶ Kyoto Encyclopedia of Genes and Genomes,³⁷ mygene.info,³⁸ myvariant.info,³⁸ ensembl,³⁹ Human Metabolome Database,⁴⁰ UniProt Knowledgebase,⁴¹ and bio2RDF.⁴² ROBOKOP KG currently contains approximately 500,000 nodes and 12 million edges. A simplified version of the database schema is provided as a graphical abstract.

2.8. Example Queries. Queries of the ROBOKOP KG can be made using the cypher query language,⁴³ which supports queries of arbitrary patterns. The ROBOKOP KG can also be queried via the ROBOKOP application user interface (UI), which provides an additional analytic algorithm that scores and ranks results.

2.8.1. Direct Cypher Queries of the ROBOKOP KG. Case Example 1: Linear Queries of Entity Relationships. Simple linear queries, with one or more interior nodes, are a common use case for ROBOKOP KG. An example “two-hop” cypher query of the ROBOKOP KG asks *what cells are part of the brain and related to the biological process of thyroid stimulating*

hormone secretion? The query itself is structured as a linear chain as follows:

anatomical entity (brain) – cell – biological process or activity (thyroid stimulating hormone secretion)

When executed, ROBOKOP KG returns a single cell type, thyrotroph. Here, the entities at either end of the chain are specified, and the query is intended to find unknown concepts that relate them.

Case Example 2: Mechanistic Identification of Potential Disease Treatments. Figure 1 provides a graphical representa-

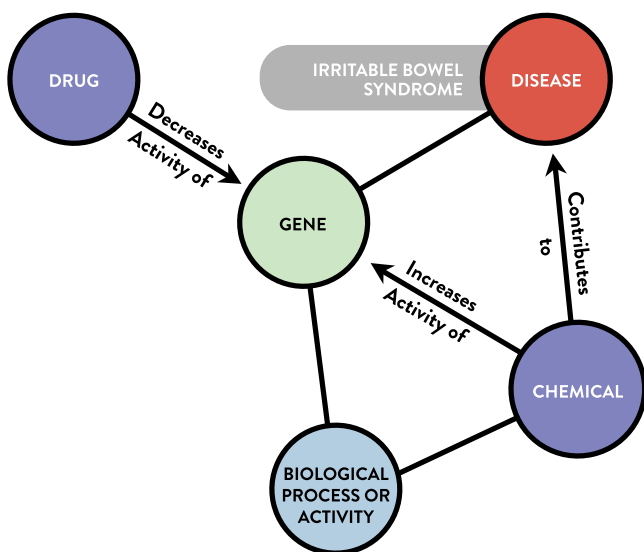


Figure 1. Graphical representation of a complex cypher query of the ROBOKOP KG. The query (available in the user guide at <http://robokopkg.renci.org>) searches for potential treatments for IBS by looking for drugs that act on disease-related genes in a manner that is opposite to that of disease-contributing chemical substances.

tion of a more complex cypher query that aims to determine potential treatments for irritable bowel syndrome (IBS) by exploiting relationships between the specified disease and contributing genes and chemical substances. Here, a user has hypothesized a relationship between chemicals that worsen IBS, disease-related intermediary genes, and chemicals that act on those genes in a manner that is opposite to that of chemicals that worsen IBS. Specifically, the query asks *what chemical substances contribute to IBS that also increase the activity of genes involved in IBS, and are there drugs that decrease the activity of those same genes (and thus may be useful in the treatment of IBS)?* In addition, the query includes a constraint in that the chemical-gene pairs must share the same biological process. While this constraint is not strictly necessary to implement the query, it was included in order to reduce the number of spurious answers. The execution of this query in ROBOKOP KG identifies serotonin as a contributing chemical, *HTR3A* as a shared gene, and serotonin receptor signaling pathway as a biological process that relates serotonin to *HTR3A*. Sixteen potential drugs are also returned. A review of the associated metadata from Aeolus²¹ indicates that one of the 16 drugs (alosetron) is an approved treatment for IBS and four drugs (6alpha-methylprednisolone, aripiprazole, clozapine, and dexamethasone) have IBS annotated as an adverse event that occurs less frequently than would occur by chance, thus indicating potential therapeutic activity.

Case Example 3: Using Common Disease to Discover Drugs for Repurposing for Rare Disease. Figure 2 provides a

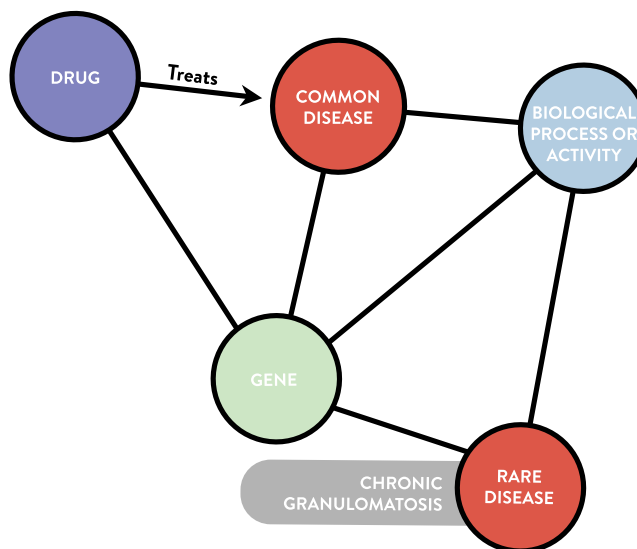


Figure 2. Graphical representation of a complex cypher query of the ROBOKOP KG. The query (available in the user guide at <http://robokopkg.renci.org>) looks for putative rare disease treatments among treatments for biologically related common diseases, i.e., common diseases that share both a gene and a biological process with the input rare disease.

graphical representation of a cypher query that aims to identify drugs that treat a common disease and are repurposing candidates for the treatment of a rare disease. Specifically, the query asks *are there drugs for common diseases that share a gene and related biological process with the rare disease chronic granulomatosis?* When the query is executed in ROBOKOP KG, we find that psoriatic arthritis is one common disease that is linked to chronic granulomatosis disease via the biological process inflammatory response and the *CYBB* gene. The drug that is returned is folic acid. A review of the associated metadata from the Comparative Toxicogenomics Database²² establishes an interaction between folic acid and methotrexate. A subsequent Google search of “folic acid” and “chronic granulomatosis” identifies several sources⁴⁴ that support the use of folic acid to reduce the side effects of methotrexate in the treatment of a subtype of chronic granulomatosis, namely, granulomatosis with polyangiitis.

2.9. ROBOKOP Queries of the ROBOKOP KG. In addition to direct cypher queries of the ROBOKOP KG, the ROBOKOP KG can be used as a data provider to a series of specific or general query UIs, including the application ROBOKOP, which applies a complex scoring and ranking algorithm to the answer subgraphs that are returned to users.⁴

Case Example 4: Mechanistic Explanations for Unexpected Clinical Observations. A recent query that was posed to ROBOKOP was driven by a real-world clinical observation. Specifically, an experimental trial of inhaled isopropyl alcohol, suggested by literature evidence,^{45,46} was found to be effective in the treatment of cyclic vomiting syndrome. The medical care team involved in this case wanted to identify the potential mechanism of action for isopropyl alcohol, so the following query was posed to ROBOKOP via the ROBOKOP UI: *why is isopropyl alcohol effective in the treatment of cyclic vomiting disorder?* The machine question was

http://robokop.renci.org/a/2c1f54ad-9ce3-476b-990b-6b41d33777ca_4fb9b390-f98d-44a8-bb3a-4c96cf7b7112/

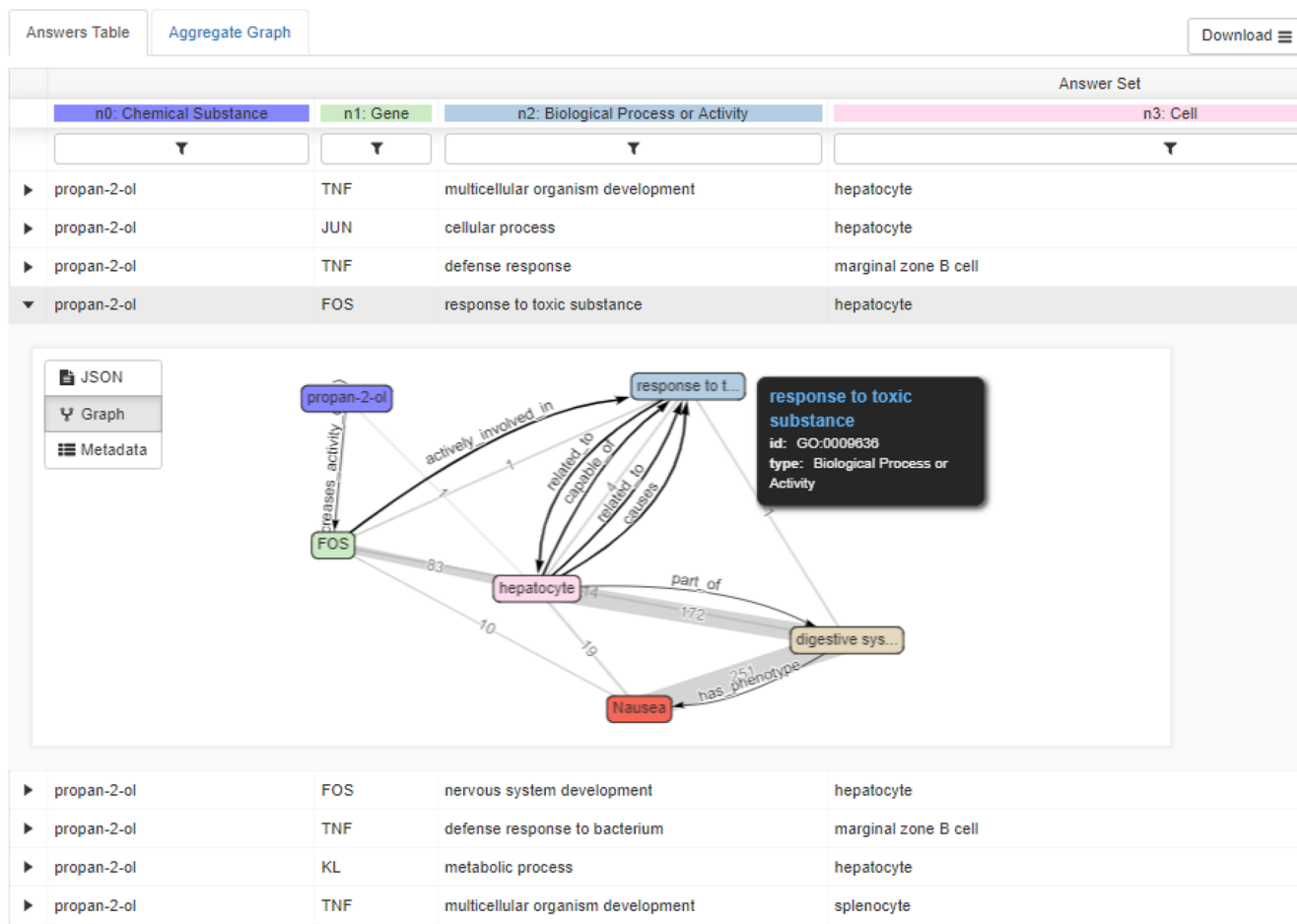


Figure 3. One of the top answer subgraphs returned by ROBOKOP in response to the following query: *isopropyl alcohol* → *gene* → *biological process or activity* → *cell* → *anatomical entity* → *nausea*. Of note, users can interact with the graph and explore the supporting publications, which can be accessed by clicking on an edge. Users can also click on an edge to determine the provenance of predicate assertions between two nodes and associated metadata or click on a node to retrieve metadata, as shown here for “response to toxic substance”.

in the form of a graph query aimed at finding a linear path that starts with isopropyl alcohol and ends with nausea, while traversing a specified set of node types in between, as follows:

isopropyl alcohol → *gene* → *biological process or activity* → *cell* → *anatomical entity* → *nausea*

Multiple ranked answer subgraphs are returned in response to the query, with each subgraph representing a different path through the ROBOKOP KG. The answer subgraphs are scored and ranked using an algorithm that factors in the number of supporting PubMed publications, as well as indirect support provided by literature co-occurrence of all pairwise sets of terms in the path.⁴ One of the top answer subgraphs for this query, shown in Figure 3, suggests that isopropyl alcohol may be acting via the *Fos* gene to alter the response of hepatocytes to toxic substances and indirectly decrease nausea. The medical care team is now actively exploring this hypothesized mechanism of action.

3. CONCLUSION

ROBOKOP is currently supporting knowledge discovery as part of the Translator program, but we anticipate that the wide range of knowledge integrated into the ROBOKOP KG will have broad applicability outside of the Translator program, as our approach of rational data integration, modularity, and well-

designed interfaces allows innumerable applications to be built from ROBOKOP components.

The modular design of ROBOKOP allows us to release the ROBOKOP KG independent of the reasoning, inference, and other algorithms that interpret the data set and are accessible via the ROBOKOP application. We are hosting a read-only version of the database at <http://robokopkg.renci.org> and regularly updating it with new data sources. We are also provisioning the full database as an export with which users can easily create their own local instance. The user guide, located at the ROBOKOP KG Web site, contains further information about the database, sample queries, similarity searches, and other information.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: bizon@renci.org.

ORCID

Karamarie Fecho: 0000-0002-6704-9306

Alexander Tropsha: 0000-0003-3802-8896

Author Contributions

The manuscript was written with contributions from all authors. All authors have given approval to the final version of the manuscript.

Notes

ROBOKOP KG, including downloadable copies and example queries, is available at <http://robokopkg.renci.org/>. ROBOKOP KGB is openly available under the MIT software license from <https://github.com/NCATS-Gamma/robokop-interfaces>.

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

This work was supported by the National Center for Advancing Translational Sciences, National Institutes of Health (Grant OT2TR002514 to A.T.). The authors thank Matt Brush and Chris Mungall for creation of the BioLink data model, Eric Deutch and David Koslicki for contributions to the Translator KG API specification, Richard Bruskewich for advice on services and useful discussions, and Matt Might and Will Byrd for feedback on the ROBOKOP KG and the isopropyl alcohol use case.

ABBREVIATIONS

API, Application Programming Interface; IBS, irritable bowel syndrome; KG, Knowledge Graph; KGB, Knowledge Graph Builder; NFA, nondeterministic finite automaton; ROBOKOP, Reasoning Over Biomedical Objects linked in Knowledge-Oriented Pathways; UI, user interface

REFERENCES

- (1) Singhal, A. Introducing the Knowledge Graph: things, not strings. <https://www.blog.google/products/search/introducing-knowledge-graph-things-not/> (accessed July 13, 2019).
- (2) Wilcke, X.; Bloem, P.; De Boer, V. The Knowledge Graph as the Default Data Model for Learning on Heterogeneous Knowledge. *Data Science* **2017**, *1* (1–2), 39–57.
- (3) Collarana, D.; Galkin, M.; Traverso-Ribon, I.; Lange, C.; Vidal, M.-E.; Auer, S. Semantic Data Integration for Knowledge Graph Construction at Query Time. In *2017 IEEE 11th International Conference on Semantic Computing (ICSC)*; IEEE, 2017; pp 109–116.
- (4) Morton, K.; Wang, P.; Bizon, C.; Cox, S.; Balhoff, J.; Kebede, Y.; Fecho, K.; Tropsha, A. ROBOKOP: An Abstraction Layer and User Interface for Knowledge Graphs to Support Question Answering. *Bioinformatics* **2019**, Aug 13, No. btz604, DOI: 10.1093/bioinformatics/btz604.
- (5) Biomedical Data Translator Consortium. Toward A Universal Biomedical Data Translator. *Clin. Transl. Sci.* **2019**, *12* (2), 86–90.
- (6) Biolink model. <https://biolink.github.io/biolink-model/> (accessed July 13, 2019).
- (7) Ashburner, M.; Ball, C. A.; Blake, J. A.; Botstein, D.; Butler, H.; Cherry, J. M.; Davis, A. P.; Dolinski, K.; Dwight, S. S.; Eppig, J. T.; et al. Gene Ontology: Tool for the Unification of Biology. *The Gene Ontology Consortium. Nat. Genet.* **2000**, *25* (1), 25–29.
- (8) Zaveri, A.; Dastgheib, S.; Wu, C.; Whetzel, T.; Verborgh, R.; Avillach, P.; Korodi, G.; Terryn, R.; Jagodnik, K.; Assis, P. smartAPI: Towards a More Intelligent Network of Web APIs. In *The Semantic Web*; Blomqvist, E., Maynard, D., Gangemi, A., Hoekstra, R., Hitzler, P., Hartig, O., Eds.; Lecture Notes in Computer Science, Vol. 10250; Springer: Cham, 2017; pp 154–169.
- (9) SmartBag, Github. <https://github.com/RishabhBrajabasi/SmartBag> (accessed December 2019).
- (10) Emonet, V.; Malic, A.; Zaveri, A.; Grigoriu, A.; Dumontier, M. Data2Services: Enabling Automated Conversion of Data to Services. *Figshare*, 2018. DOI: 10.6084/m9.figshare.7345868.v1.

- (11) Chen, B.; Dong, X.; Jiao, D.; Wang, H.; Zhu, Q.; Ding, Y.; Wild, D. J. Chem2Bio2RDF: A Semantic Framework for Linking and Data Mining Chemogenomic and Systems Chemical Biology Data. *BMC Bioinf.* **2010**, *11*, 255.
- (12) Himmelstein, D. S.; Baranzini, S. E. Heterogeneous Network Edge Prediction: A Data Integration Approach to Prioritize Disease-Associated Genes. *PLoS Comput. Biol.* **2015**, *11* (7), No. e1004259.
- (13) Womack, F.; McClelland, J.; Koslicki, D. Leveraging Distributed Biomedical Knowledge Sources to Discover Novel Uses for Known Drugs. *bioRxiv* **2019**, 765305.
- (14) Povey, S.; Lovering, R.; Bruford, E.; Wright, M.; Lush, M.; Wain, H. The HUGO Gene Nomenclature Committee (HGNC). *Hum. Genet.* **2001**, *109* (6), 678–680.
- (15) Mondo Disease Ontology. <http://www.obofoundry.org/ontology/mondo.html> (accessed July 13, 2019).
- (16) Chambers, J.; Davies, M.; Gaulton, A.; Hersey, A.; Velankar, S.; Petryszak, R.; Hastings, J.; Bellis, L.; McGlinchey, S.; Overington, J. P. UniChem: A Unified Chemical Structure Cross-Referencing and Identifier Tracking System. *J. Cheminf.* **2013**, *5* (1), 3.
- (17) Medical Subject Headings, 2019. NIH. <https://www.nlm.nih.gov/mesh/meshhome.html> (accessed December 2019).
- (18) ICD-11. WHO. <https://icd.who.int/en> (accessed November 21, 2019).
- (19) Wishart, D. S.; Feunang, Y. D.; Guo, A. C.; Lo, E. J.; Marcu, A.; Grant, J. R.; Sajed, T.; Johnson, D.; Li, C.; Sayeeda, Z.; et al. DrugBank 5.0: A Major Update to the Drug Bank Database for 2018. *Nucleic Acids Res.* **2018**, *46* (D1), D1074–D1082.
- (20) Ursu, O.; Holmes, J.; Knockel, J.; Bologa, C. G.; Yang, J. J.; Mathias, S. L.; Nelson, S. J.; Oprea, T. I. Drug Central: Online Drug Compendium. *Nucleic Acids Res.* **2017**, *45* (D1), D932–D939.
- (21) Banda, J. M.; Evans, L.; Vanguri, R. S.; Tatonetti, N. P.; Ryan, P. B.; Shah, N. H. A Curated and Standardized Adverse Drug Event Resource to Accelerate Drug Safety Research. *Sci. Data* **2016**, *3*, 160026.
- (22) Davis, A. P.; Grondin, C. J.; Johnson, R. J.; Sciaky, D.; McMorran, R.; Wieggers, J.; Wieggers, T. C.; Mattingly, C. J. The Comparative Toxicogenomics Database: Update 2019. *Nucleic Acids Res.* **2019**, *47* (D1), D948–D954.
- (23) Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; et al. PubChem 2019 Update: Improved Access to Chemical Data. *Nucleic Acids Res.* **2019**, *47* (D1), D1102–D1109.
- (24) Mi, H.; Muruganujan, A.; Ebert, D.; Huang, X.; Thomas, P. D. PANTHER Version 14: More Genomes, a New PANTHER GO-Slim and Improvements in Enrichment Analysis Tools. *Nucleic Acids Res.* **2019**, *47* (D1), D419–D426.
- (25) Gaulton, A.; Hersey, A.; Nowotka, M.; Bento, A. P.; Chambers, J.; Mendez, D.; Mutowo, P.; Atkinson, F.; Bellis, L. J.; Cibrián-Uhalte, E.; et al. The ChEMBL Database in 2017. *Nucleic Acids Res.* **2017**, *45* (D1), D945–D954.
- (26) Hastings, J.; Owen, G.; Dekker, A.; Ennis, M.; Kale, N.; Muthukrishnan, V.; Turner, S.; Swainston, N.; Mendes, P.; Steinbeck, C. ChEBI in 2016: Improved Services and an Expanding Collection of Metabolites. *Nucleic Acids Res.* **2016**, *44* (D1), D1214–D1219.
- (27) The same high-performance Bio Things API for chemicals and drugs. <https://mychem.info/> (accessed July 13, 2019).
- (28) Biolink-Api. Github. <https://github.com/biolink/biolink-api> (accessed December 2019).
- (29) Köhler, S.; Carmody, L.; Vasilevsky, N.; Jacobsen, J. O. B.; Danis, D.; Gouridine, J.-P.; Gargano, M.; Harris, N. L.; Matentzoglou, N.; McMurry, J. A.; et al. Expansion of the Human Phenotype Ontology (HPO) Knowledge Base and Resources. *Nucleic Acids Res.* **2019**, *47* (D1), D1018–D1027.
- (30) The Gene Ontology Consortium. The Gene Ontology Resource: 20 Years and Still GOing Strong. *Nucleic Acids Res.* **2019**, *47* (D1), D330–D338.
- (31) Binns, D.; Dimmer, E.; Huntley, R.; Barrell, D.; O'Donovan, C.; Apweiler, R. Quick GO: A Web-Based Tool for Gene Ontology Searching. *Bioinformatics* **2009**, *25* (22), 3045–3046.

- (32) Carbon, S.; Ireland, A.; Mungall, C. J.; Shu, S.; Marshall, B.; Lewis, S. AmiGO Hub; Web Presence Working Group. AmiGO: Online Access to Ontology and Annotation Data. *Bioinformatics* **2009**, *25* (2), 288–289.
- (33) Nguyen, D.-T.; Mathias, S.; Bologa, C.; Brunak, S.; Fernandez, N.; Gaulton, A.; Hersey, A.; Holmes, J.; Jensen, L. J.; Karlsson, A.; et al. Pharos: Collating Protein Information to Shed Light on the Druggable Genome. *Nucleic Acids Res.* **2017**, *45* (D1), D995–D1002.
- (34) Pawliczek, P.; Patel, R. Y.; Ashmore, L. R.; Jackson, A. R.; Bizon, C.; Nelson, T.; Powell, B.; Freimuth, R. R.; Strande, N.; Shah, N.; et al. ClinGen Allele Registry Links Information about Genetic Variants. *Hum. Mutat.* **2018**, *39* (11), 1690–1701.
- (35) Landrum, M. J.; Lee, J. M.; Benson, M.; Brown, G. R.; Chao, C.; Chitipiralla, S.; Gu, B.; Hart, J.; Hoffman, D.; Jang, W.; et al. ClinVar: Improving Access to Variant Interpretations and Supporting Evidence. *Nucleic Acids Res.* **2018**, *46* (D1), D1062–D1067.
- (36) Buniello, A.; MacArthur, J. A. L.; Cerezo, M.; Harris, L. W.; Hayhurst, J.; Malangone, C.; McMahon, A.; Morales, J.; Mountjoy, E.; Sollis, E.; et al. The NHGRI-EBI GWAS Catalog of Published Genome-Wide Association Studies, Targeted Arrays and Summary Statistics 2019. *Nucleic Acids Res.* **2019**, *47* (D1), D1005–D1012.
- (37) Kanehisa, M.; Goto, S. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.* **2000**, *28* (1), 27–30.
- (38) Xin, J.; Mark, A.; Afrasiabi, C.; Tsueng, G.; Juchler, M.; Gopal, N.; Stupp, G. S.; Putman, T. E.; Ainscough, B. J.; Griffith, O. L.; et al. High-Performance Web Services for Querying Gene and Variant Annotation. *Genome Biol.* **2016**, *17* (1), 91.
- (39) Zerbino, D. R.; Achuthan, P.; Akanni, W.; Amode, M. R.; Barrell, D.; Bhai, J.; Billis, K.; Cummins, C.; Gall, A.; Girón, C. G.; et al. Ensembl 2018. *Nucleic Acids Res.* **2018**, *46* (D1), D754–D761.
- (40) Wishart, D. S.; Tzur, D.; Knox, C.; Eisner, R.; Guo, A. C.; Young, N.; Cheng, D.; Jewell, K.; Arndt, D.; Sawhney, S.; et al. HMDB: The Human Metabolome Database. *Nucleic Acids Res.* **2007**, *35* (Database), D521–D526.
- (41) UniProt Consortium. UniProt: A Worldwide Hub of Protein Knowledge. *Nucleic Acids Res.* **2019**, *47* (D1), D506–D515.
- (42) Callahan, A.; Cruz-Toledo, J.; Ansell, P.; Dumontier, M. Bio2RDF Release 2: Improved Coverage, Interoperability and Provenance of Life Science Linked Data. In *The Semantic Web: Semantics and Big Data*; Springer: Berlin Heidelberg, 2013; pp 200–212.
- (43) Cypher Query Language Developer Guides & Tutorials. <http://neo4j.com/developer/cypher-query-language/> (accessed July 13, 2019).
- (44) Granulomatosis with polyangiitis (GPA). <https://www.arthritis.org/about-arthritis/types/granulomatosis-with-polyangiitis/> (accessed August 13, 2019).
- (45) Wang, S. M.; Hofstadter, M. B.; Kain, Z. N. An Alternative Method to Alleviate Postoperative Nausea and Vomiting in Children. *J. Clin. Anesth.* **1999**, *11* (3), 231–234.
- (46) Pellegrini, J.; DeLoge, J.; Bennett, J.; Kelly, J. Comparison of Inhalation of Isopropyl Alcohol vs Promethazine in the Treatment of Postoperative Nausea and Vomiting (PONV) in Patients Identified as at High Risk for Developing PONV. *AANA J.* **2009**, *77* (4), 293–299.