Phenotype Knowledge Translator: A FAIR Ecosystem for Representing Large-Scale Biomedical Knowledge

SUPPLEMENTAL MATERIAL

This section provides additional figures and tables provided to help explain algorithm dependencies, resources used for evaluation, and the results from performing an open-source software survey.

Table of Contents

Supplemental Table 1. Definitions and Acronyms.

Concept/Acronym	Definition
Edge, Relation	Observed connections between nodes. Edges or relations are used to specify different types of relationships (e.g., interaction, substance that treats) that can exist between a pair of nodes.
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 can infer new knowledge to address a variety of applications and use cases.
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.
Triple	A node-relation-node statement (e.g., geneA - interacts with - geneB).

Note. Concepts are ordered alphabetically.

Supplemental Figure 1. PheKnowLator Data Download Dependencies.

A detailed description of the `resource_info.txt`, `ontology_source_list.txt`, and `edge_source_list.txt` input files are provided. These are the three primary input files required to construct PheKnowLator knowledge graphs. Additional information is provided on GitHub (https://github.com/callahantiff/PheKnowLator/wiki/Dependencies).

==========
Tue Jul 06 17:00:00 UTC 2021
==========

DATA INFO
  - DOWNLOAD_URL = http://purl.obolibrary.org/obo/hp.owl
  - DOWNLOAD_DATE = 07/06/2021
  - FILE_SIZE_IN_BYTES = 80747939
  - GOOGLE_CLOUD_STORAGE_URL = https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_06JUL2021/data/original_data/hp_with_imports.owl
DATA INFO
  - DOWNLOAD_URL = http://purl.obolibrary.org/obo/go.owl
  - DOWNLOAD_DATE = 07/06/2021
  - FILE_SIZE_IN_BYTES = 132079457
  - GOOGLE_CLOUD_STORAGE_URL = https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_06JUL2021/data/original_data/go_with_imports.owl
DATA INFO
  - DOWNLOAD_URL = http://purl.obolibrary.org/obo/mondo.owl
  - DOWNLOAD_DATE = 07/06/2021
  - FILE_SIZE_IN_BYTES = 230262679
  - GOOGLE_CLOUD_STORAGE_URL = https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_06JUL2021/data/original_data/mondo_with_imports.owl
DATA INFO
  - DOWNLOAD_URL = http://purl.obolibrary.org/obo/vo.owl
  - DOWNLOAD_DATE = 07/06/2021
  - FILE_SIZE_IN_BYTES = 8020499
  - GOOGLE_CLOUD_STORAGE_URL = https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_06JUL2021/data/original_data/vo_with_imports.owl
DATA INFO
  - DOWNLOAD_URL = ftp://ftp.ebi.ac.uk/pub/databases/genenames/new/tsv/hgnc_complete_set.txt
  - DOWNLOAD_DATE = 07/06/2021
  - FILE_SIZE_IN_BYTES = 15874757
  - GOOGLE_CLOUD_STORAGE_URL = https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_06JUL2021/data/original_data/hgnc_complete_set.txt
DATA INFO
  - DOWNLOAD_URL = https://proconsortium.org/download/current/promapping.txt
  - DOWNLOAD_DATE = 07/06/2021
  - FILE_SIZE_IN_BYTES = 15271039
  - GOOGLE_CLOUD_STORAGE_URL = https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_06JUL2021/data/original_data/promapping.txt

Supplemental Figure 2. PheKnowLator Data Download Metadata.
An example of the metadata generated for all downloaded data sources and saved to a file called `downloaded_build_metadata.txt` for each build.

```
================================================
ONTOLOGY CLEANING REPORT
Sun May 02 07:02:05 UTC 2021
================================================

INDIVIDUAL ONTOLOGY: vo_with_imports.owl
**************************************************
    - Original GCS URL: https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_01MAY2021/data/original_data/vo_with_imports.owl
    - Processed GCS URL: https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_01MAY2021/data/processed_data/vo_with_imports.owl
    - Statistics Before Cleaning:
        - 86454 Triples; 7089 Classes; 165 Individuals; 232 Object Properties; 97 Annotation Properties; 5 Connected Components
    - Statistics After Cleaning:
        - 89764 Triples; 7085 Classes; 165 Individuals; 232 Object Properties; 97 Annotation Properties; 5 Connected Components
    - Value Errors: 0
    - Identifier Errors (n=2):
        - http://purl.obolibrary.org/obo/PRO_000000001
        - http://purl.obolibrary.org/obo/PRO_000015399
    - Deprecated Classes: 0
    - Obsolete Classes: 0
    - Punning Errors:
        - Classes: 0
        - Object Properties: 0


MERGED ONTOLOGY: PheKnowLator_MergedOntologies.owl
**************************************************
    - Original GCS URL: https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_01MAY2021/data/original_data/PheKnowLator_MergedOntologies.owl
    - Processed GCS URL: https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_01MAY2021/data/processed_data/PheKnowLator_MergedOntologies.owl
    - Statistics Before Cleaning:
        - 13746883 Triples; 548947 Classes; 195 Individuals; 847 Object Properties; 656 Annotation Properties; 8 Connected Components
    - Statistics After Cleaning:
        - 13748009 Triples; 545259 Classes; 188 Individuals; 846 Object Properties; 656 Annotation Properties; 8 Connected Components
    - Value Errors: 0
    - Identifier Errors (n=2):
        - http://purl.obolibrary.org/obo/PRO_000000001
        - http://purl.obolibrary.org/obo/PRO_000015399
    - Punning Errors:
        - Classes (n=8):
            - http://purl.obolibrary.org/obo/NCBITaxon_6040
            - http://purl.obolibrary.org/obo/NCBITaxon_41324
            - http://purl.obolibrary.org/obo/NCBITaxon_147099
            - http://purl.obolibrary.org/obo/NCBITaxon_6073
            - http://purl.obolibrary.org/obo/NCBITaxon_6157
            - http://purl.obolibrary.org/obo/NCBITaxon_110815
            - http://purl.obolibrary.org/obo/CLO_0054409
            - http://purl.obolibrary.org/obo/NCBITaxon_8570
        - Object Properties: 0
    - Semantic Heterogeneity:
        - Normalized Entities (n=7):
            - OGG_0000000002 rdfs:subClassOf SO_0000704
            - PR_000000001 rdfs:subClassOf SO_0000104
            - CHEBI_36080 rdfs:subClassOf SO_0000104
            - OGMS_0000045 rdfs:subClassOf MONDO_0000001
            - OBI_1110034 rdfs:subClassOf CHEBI_59132
            - VO_0003030 rdfs:subClassOf CHEBI_5291
            - FMA_12278 rdfs:subClassOf CHEBI_24621
        - Aligned HGNC IDs: 23624
        - Other Classes that May Need Alignment: 391041
        - Deprecated Ontology HGNC Identifiers Needing Alignment: 0
```

Supplemental Figure 3. Ontology Data Quality Report Output.

Example output from the ontology data quality report generated as part of the ontology preprocessing pipeline in the PheKnowLator Ecosystem. This snippet was obtained from the v2.1.0 01MAY2021 build report (`ontology cleaning report.txt`). Each report is time-stamped and contains the following information for each individual ontology and the set of merged ontologies: (i) Google Cloud Storage Bucket location for the downloaded ontologies; (ii) Google Cloud Storage Bucket location for the cleaned ontologies; (iii) descriptive statistics (i.e., triples, classes, named individuals, object properties, annotation properties, and connected components counts) before and after applying the DQ checks; and (iv) the count of identified errors for each DQ check. The URI for each identified error for all the checks, except Semantic Heterogeneity and Identifier Alignment, are also provided.

==========
Wed Jul 07 00:28:07 UTC 2021
==========

DATA INFO
- DOWNLOAD_URL = https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_06JUL2021/data/original_data/CLINVAR_VARIANT_GENE_DISEASE_PHENOTYPE_EDGES.txt
- DOWNLOAD_DATE = 07/07/2021
- FILE_SIZE_IN_BYTES = 3441280758
- GOOGLE_CLOUD_STORAGE_URL = https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_06JUL2021/data/processed_data/CLINVAR_VARIANT_GENE_DISEASE_PHENOTYPE_EDGES.txt

DATA INFO
- DOWNLOAD_URL = https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_06JUL2021/data/original_data/DISEASE_MONDO_MAP.txt
- DOWNLOAD_DATE = 07/07/2021
- FILE_SIZE_IN_BYTES = 3691756
- GOOGLE_CLOUD_STORAGE_URL = https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_06JUL2021/data/processed_data/DISEASE_MONDO_MAP.txt

DATA INFO
- DOWNLOAD_URL = https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_06JUL2021/data/original_data/ENSEMBL_GENE_ENTREZ_GENE_MAP.txt
- DOWNLOAD_DATE = 07/07/2021
- FILE_SIZE_IN_BYTES = 3552240
- GOOGLE_CLOUD_STORAGE_URL = https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_06JUL2021/data/processed_data/ENSEMBL_GENE_ENTREZ_GENE_MAP.txt

DATA INFO
- DOWNLOAD_URL = https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_06JUL2021/data/original_data/ENSEMBL_TRANSCRIPT_PROTEIN_ONTOLOGY_MAP.txt
- DOWNLOAD_DATE = 07/07/2021
- FILE_SIZE_IN_BYTES = 2931638
- GOOGLE_CLOUD_STORAGE_URL = https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_06JUL2021/data/processed_data/ENSEMBL_TRANSCRIPT_PROTEIN_ONTOLOGY_MAP.txt

Supplemental Figure 4. PheKnowLator Data Preparation Metadata.

An example of the metadata output generated for all processed data sources. These data are output for all data that are preprocessed as part building each knowledge graph. These data are saved to a file called `preprocessed_build_metadata.txt`.

==========
#Wed Jul 07 01:15:25 UTC 2021
==========

EDGE: chemical-disease
DATA PROCESSING INFO
  - IDENTIFIER MAPPING = chemical (./resources/processed_data/MESH_CHEBI_MAP.txt) | disease (./resources/processed_data/DISEASE_MONDO_MAP.txt)
  - FILTERING CRITERIA = None
  - EVIDENCE CRITERIA = data[5]!=''
DATA INFO
  - DOWNLOAD_URL = https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_06JUL2021/data/original_data/CTD_chemicals_diseases.tsv
  - DOWNLOAD_DATE = 07/07/2021
  - FILE_SIZE_IN_BYTES = 691470262
  - DOWNLOADED_FILE_LOCATION = resources/edge_data/chemical-disease_CTD_chemicals_diseases.tsv

EDGE: chemical-gene
DATA PROCESSING INFO
  - IDENTIFIER MAPPING = chemical (./resources/processed_data/MESH_CHEBI_MAP.txt)
  - FILTERING CRITERIA = data[6]==Homo sapiens | data[5].startswith('gene')
  - EVIDENCE CRITERIA = data[9]affectsnot in x
DATA INFO
  - DOWNLOAD_URL = https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_06JUL2021/data/original_data/CTD_chem_gene_ixns.tsv
  - DOWNLOAD_DATE = 07/07/2021
  - FILE_SIZE_IN_BYTES = 433146998
  - DOWNLOADED_FILE_LOCATION = resources/edge_data/chemical-gene_CTD_chem_gene_ixns.tsv

EDGE: chemical-gobp
DATA PROCESSING INFO
  - IDENTIFIER MAPPING = chemical (./resources/processed_data/MESH_CHEBI_MAP.txt)
  - FILTERING CRITERIA = data[3]==Biological Process
  - EVIDENCE CRITERIA = data[8]<=1.04e-47
DATA INFO
  - DOWNLOAD_URL = https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_06JUL2021/data/original_data/CTD_chem_go_enriched.tsv
  - DOWNLOAD_DATE = 07/07/2021
  - FILE_SIZE_IN_BYTES = 834396980
  - DOWNLOADED_FILE_LOCATION = resources/edge_data/chemical-gobp_CTD_chem_go_enriched.tsv

Supplemental Figure 5. PheKnowLator Edge Source Metadata.

This figure provides an example of the metadata generated for all non-ontology data sources, which are saved to a file called `edge_source_metadata.txt` for each build.

==========
#Wed Jul 07 01:14:26 UTC 2021
==========

EDGE: phenotype
DATA PROCESSING INFO
  - IDENTIFIER MAPPING = None
  - FILTERING CRITERIA = None
  - EVIDENCE CRITERIA = None
DATA INFO
  - DOWNLOAD_URL = https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_06JUL2021/data/processed_data/hp_with_imports.owl
  - DOWNLOAD_DATE = 07/07/2021
  - FILE_SIZE_IN_BYTES = 80761144
  - DOWNLOADED_FILE_LOCATION = resources/ontologies/hp_with_imports_with_imports.owl

EDGE: go
DATA PROCESSING INFO
  - IDENTIFIER MAPPING = None
  - FILTERING CRITERIA = None
  - EVIDENCE CRITERIA = None
DATA INFO
  - DOWNLOAD_URL = https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_06JUL2021/data/processed_data/go_with_imports.owl
  - DOWNLOAD_DATE = 07/07/2021
  - FILE_SIZE_IN_BYTES = 122816805
  - DOWNLOADED_FILE_LOCATION = resources/ontologies/go_with_imports_with_imports.owl

EDGE: disease
DATA PROCESSING INFO
  - IDENTIFIER MAPPING = None
  - FILTERING CRITERIA = None
  - EVIDENCE CRITERIA = None
DATA INFO
  - DOWNLOAD_URL = https://storage.googleapis.com/pheknowlator/archived_builds/release_v2.1.0/build_06JUL2021/data/processed_data/mondo_with_imports.owl
  - DOWNLOAD_DATE = 07/07/2021
  - FILE_SIZE_IN_BYTES = 226235238
  - DOWNLOADED_FILE_LOCATION = resources/ontologies/mondo_with_imports_with_imports.owl

Supplemental Figure 6. PheKnowLator Ontology Source Metadata.

This figure provides an example of the metadata generated for all downloaded ontologies, which are saved to a file called `ontology_source_metadata.txt` for each build.

Supplemental Table 2. PheKnowLator Knowledge Graph Output.

File	File Description
*_MergedOntologies.owl	This RDF/XML formatted file only contains the baseline set of cleaned merged ontologies.
*_OWL_LogicOnly.nt	This N-Triples formatted file contains the logical axioms for the baseline set of cleaned merged ontologies and all non-ontology edges. It does not contain any annotation assertions (i.e., metadata like labels, definitions, and synonyms). This file contains the minimum logical subset needed to run a description logic reasoner.
*_OWL_AnnotationsOnly.nt	This N-Triples formatted file contains annotation assertions (i.e., metadata like labels, definitions, and synonyms) for the baseline set of cleaned merged ontologies and all non-ontology edges.
*_OWL.nt	This N-Triples formatted file contains the baseline set of cleaned merged ontologies and all non-ontology edges. It also contains all annotation assertions (i.e., metadata like labels, definitions, and synonyms). This file contains all OWL semantics needed to run a description logic reasoner.
OWL-NETS Base The OWL-NETS files have undergone a transformation that decodes all OWL semantics in order to create a graph that only contains biologically relevant nodes and edges and is much more useful for inductive types of machine learning. For more information see: https://github.com/callahantiff/PheKnowLator/wiki/OWL-NETS-2.0.	
*_noOWL_OWLNETS.nt	This N-Triples formatted file contains the OWL-NETS transformed build (the build that's transformed is the build stored in the `*_inverseRelations_OWL.owl` file).
*_noOWL_OWLNETS_NetworkxMultiDiGraph.gpickle	This file is a NetworkX MultiDiGraph representation of the same content that is stored in the `*_inverseRelations_noOWL_OWLNETS.nt` file. Note that this representation includes keys for nodes and edges (node: key = URI; edge: predicate_key = MD5 hash of triple "s_uri" + "p_uri" + "o_uri"). Each edge also has a default weight of 0.0.
*_noOWL_OWLNETS_decoding_dict.pkl	This dictionary stores details about the OWL-NETS transformation. Specifically, it contains metadata that can be used to reverse the transformation.
*_noOWL_Triples_Identifiers .txt	This tab-delimited text file contains the same information as the.nt and gpickle files but is organized into a common format used for many graph representational learning algorithms. The file contains three columns, one for each part of a triple (i.e., subject, predicate, object), where each identifier is the full resolvable URI.
*_noOWL_Triples_Integers.txt	This tab-delimited text file contains the same information as the.nt and.gpickle files, but is organized into a common format used for many graph representational learning algorithms. The file contains three columns, one for each part of a triple (i.e., subject, predicate, object), where each identifier is the full resolvable URI. The primary difference between this file and the `_idenitfiers` file is that the identifier URIs have been mapped to integers.
*_noOWL_Triples_Integer_Identifier_Map.json	This JSON file contains a dictionary where the keys are node identifiers, and the values are integers. It converts the `_indentifiers` file to the `_integers` file.

File	File Description
*_noOWL_NodeLabels.txt	This tab-delimited .txt file contains metadata on all nodes and relations in the N-Triples, gpickle, and _identifiers files. It contains the following columns: 1) entity_type (e.g., "NODES", "RELATIONS", or "NA" if not a `owl:Class`, `owl:NamedIndividual`, `owl:ObjectProperty` or `owl:AnnotationProperty`); 2) integer_id (e.g., 1 - the integer used to represent this URI in the Edge List output -- matches the integer assignment from the `_integers` file); 3) entity_uri (e.g., "GO_0048252"); 4) label (e.g., "lauric acid metabolic process"); 5) description/definition (e.g., "The chemical reactions and pathways involving lauric acid, a fatty acid with the formula CH3(CH2)10COOH. Derived from vegetable sources."); and 6) synonym (e.g., "lauric acid metabolism|n-dodecanoic acid metabolic process |n-dodecanoic acid metabolism") NOTE. For all non-OWL-NETS builds, there will be entries in this file that contain values of NA for the entity_type column. This is expected for these types of builds; a value of NA is used for all nodes and relations that are not an `owl:Class`, `owl:NamedIndividual`, `owl:ObjectProperty` or `owl:AnnotationProperty`.
Purified OWL-NETS The purified version of an OWL-NETS build is designed to convert the base OWL-NETS build into a version that is completely consistent with a specific construction approach. For example, if the build is instance-based, then all `rdfs:subClassOf` relations are converted to `rdf:type` and for all triples where an `rdfs:subClassOf` relation occurred we add rdf:type relations between the object of this triple and all of its ancestors. For a subclass-based build, the same procedure is implemented but all occurrences of `rdf:type` are replaced with `rdfs:subClassOf`. Please note that these build types are considered experimental.	
*_noOWL_<>_purified_OWLNETS.nt	This N-Triples formatted file contains the purified OWL-NETS transformed build (the build that's transformed is the build stored in the `*_OWL.owl)file`.
*_noOWL_<>_purified_OWLNETS_NetworkxMultiDiGraph .gpickle	This file is a NetworkX MultiDiGraph representation of the same content that is stored in the `*_inverseRelations_noOWL_OWLNETS.nt` file. Note that this representation includes keys for nodes and edges (node: key = URI; edge: predicate_key = MD5 hash of triple "s_uri" + "p_uri" + "o_uri"). Each edge also has a default weight of 0.0.
*_noOWL_<>_purified_OWLNETS_decoding_dict.pkl	This dictionary stores details about the purified OWL-NETS transformation. Specifically, it contains metadata that can be used to reverse the transformation.
*_noOWL_<>_purified_Triples_Identifiers.txt	This tab-delimited text file contains the same information as the.nt and.gpickle files, but is organized into a common format used for many graph representational learning algorithms. The file contains three columns, one for each part of a triple (i.e., subject, predicate, object), where each identifier is the full resolvable URI.
*_noOWL_<>_purified_Triples_Integers.txt	This tab-delimited text file contains the same information as the nt and.gpickle files, but is organized into a common format used for many graph representational learning algorithms. The file contains three columns, one for each part of a triple (i.e., subject, predicate, object), where each identifier is the full resolvable URI. The primary difference between this file and the `_idenitfiers` file is that the identifier URIs have been mapped to integers.
*_noOWL_<>_purified_Triples_Integer_Identifier_Map.json	This JSON file contains a dictionary where the keys are node identifiers, and the values are integers. It converts the `_identifiers` file to the `_integers` file.

File	File Description
*_noOWL_<>_purified_NodeLabels.txt	This tab-delimited .txt file contains metadata on all nodes and relations in the N-Triples, gpickle, and _identifiers files. It contains the following columns: 1) entity_type (e.g., "NODES", "RELATIONS", or "NA" if not a `owl:Class`, `owl:NamedIndividual`, `owl:ObjectProperty`, or `owl:AnnotationProperty`) 2) integer_id (e.g., 1 - the integer used to represent this URI in the Edge List output -- matches the integer assignment from the `_integers` file) 3) entity_uri (e.g., "GO_0048252") 4) label (e.g., "lauric acid metabolic process") 5) description/definition (e.g., "The chemical reactions and pathways involving lauric acid, a fatty acid with the formula CH3(CH2)10COOH. Derived from vegetable sources.") 6) synonym (e.g., "lauric acid metabolism|n-dodecanoic acid metabolic process|n-dodecanoic acid metabolism") NOTE. For all non-OWL-NETS builds, there will be entries in this file that contain values of NA for the entity_type column. This is expected for these types of builds; a value of NA is used for all nodes and relations that are not an `owl:Class`, `owl:NamedIndividual`, `owl:ObjectProperty` or `owl:AnnotationProperty`.

*is used in lieu of a particular build's specific file name.

Supplemental Table 3. Open-Source Survey Methods.

Survey date	Method	Associated GitHub repository	Publication DOI	Primary goal or objective (from GitHub)	Method validation	Most recent repository interaction
11/30/2021	Bio2BEL	https://github.com/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
6/10/2020	Bio2RDF	https://github.com/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	Examine impact of four transcription factors in Parkinson's disease	Within the last week
6/11/2020	Bio4J	https://github.com/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
6/10/2020	BioGrakn	https://github.com/graknlabs/biograkn	10.1007/978-3-319-61566-0_28	We introduce BioGrakn, 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
6/12/2020	Clinical Knowledge Graph (CKG)	https://github.com/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
6/14/2020	COVID-19-Community	https://github.com/covid-19-net/covid-19-community	NA	A community effort to build a Neo4j knowledge graph that links heterogenous data about COVID-19	Tool use demonstration	Within the last week
6/14/2020	Dipper	https://github.com/monarch-initiative/dipper	NA	Dipper is a Python package to generate RDF triples from common scientific resources.	Tool use demonstration	Within the last week
6/14/2020	Hetionet	https://github.com/hetio/hetionet	10.7554/eLife.26726	Hetionet is a hetnet — network with multiple node and edge (relationship) types — which encodes biology. The hetnet was designed for Project Rephetio	Predicted the probability of treatment for 209,168 compound–disease pairs	Within the last year
6/12/2020	iASiS Open Data Graph	https://github.com/tasosnent/Biomedical-Knowledge-Integration	arXiv:1912.08633	Propose a framework to automatically retrieve and integrate disease-specific knowledge into an up-to-date semantic graph, the iASiS Open Data Graph	Demonstrates functionality by examining use with lung cancer, dementia, and Duchenne Muscular Dystrophy	Within the last 6 months

Survey date	Method	Associated GitHub repository	Publication DOI	Primary goal or objective (from GitHub)	Method validation	Most recent repository interaction
6/14/2020	KG-COVID-19	https://github.com/Knowledge-Graph-Hub/kg-covid-19	NA	Created KG-COVID-19, a flexible framework to ingest, integrate, and remix biomedical data to produce KGs for COVID-19 response. This KG 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
6/14/2020	Knowledge Base Of Biomedicine (KaBOB)	https://github.com/UCDenver-ccp/kabob/tree/bg-integration	10.1186/s12859-015-0559-3	Present five processes for semantic data integration that, when applied collectively, solve seven key problems. These processes include 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. We demonstrate these processes and solutions by presenting KaBOB, a knowledge base of semantically integrated data	Human KG Multi-Species KG	Within the last year
6/14/2020	Knowledge Graph Exchange (KGX)	https://github.com/NCATS-Tangerine/kgx	NA	Develop 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
6/11/2020	Knowledge Graph Toolkit (KGTK)	https://github.com/usc-isi-i2/kgtk/	arXiv:2006.00088	Present KGTK, 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	Demonstrate functionality using Wikidata, DBpedia, and ConceptNet	Within the last week
6/10/2020	PheKnowLator	https://github.com/callahantiff/PheKnowLator	10.1101/2020.04.30.071407	Introduce PheKnowLator (Phenotype Knowledge Translator), a novel framework and fully automated Python 3 library explicitly designed for optimized construction of semantically-rich, large-scale biomedical knowledge graphs	Human disease mechanisms KG built and used to generate 8 KGs, which are embedded and examined using 5 different evaluation tasks	Within the last week
6/12/2020	ProNet	https://github.com/cran/ProNet	NA	Provide functions for biological network construction, visualization and analyses, including topological statistics, functional module clustering and GO-profiling	H1N1 IAV-human protein-protein interactions	> 1 year
6/14/2020	SEmantic Modeling machIne (SEMi)	https://github.com/giuseppefutia/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	Validate the software using advertising data	Within the last 6 months

Supplemental Table 4. Open-Source Survey - Knowledge Graph Construction Functionality.

Method	Data Download Functionality	Edge list Construction Functionality	KG Construction Functionality	Constructs Multiple KG Types	Other KG Construction Functionality	Process Ontology Data	Process Linked Open Data	Process Experimental Data	Process Clinical Data	Other Data Types	Data Source Processing Limit
Bio2BEL	Yes	Yes	Yes	Yes	The entire PyBEL ecosystem for I/O, conversion, database loading/querying, qualitative analysis, and quantitative analysis are all directly available for all graphs generated by Bio2BEL.	Yes	Yes	Yes	Yes	BEL can handle associative, correlative, and causal data coming from both experimental, clinical, and prior knowledge.	Not specified
Bio2RDF	Yes	Yes	Yes	Yes	Talend RESTful API Community ontology mappings SPARQL query repository 35 dataset repository	Yes	Yes	Yes	Yes	Text	Not specified
Bio4J	Yes	Yes	Yes	Yes	Titan, Angulillos API	Yes	Yes	No	No		Not specified
BioGrakn	No	No	Yes	Yes	Provides different types of API clients (Java, Python, Node.js) and a Grakn Workbase	No	Yes	Yes	Yes	Text	Not specified
Clinical Knowledge Graph (CKG)	Yes	No	Yes	No	Data preparation (filtering, imputation, formatting, normalization); Data exploration (provide summary stats); Data analysis (dimensionality reduction, visualization, simple hypothesis testing)	Yes	Yes	Yes	No		Not specified
COVID-19-Community	Yes	Yes	Yes	No	Neo4J Browser	No	Yes	No	Yes	Text, Geographic data, Census data	Not specified
Dipper	Yes	Yes	Yes	Yes	SciGraph RESTful API Build KGs with evidence and provenance	Yes	Yes	Yes	No		Not specified
Hetionet	Yes	No	Yes	No	Neo4J Browser Creates permuted KGs	Yes	Yes	Yes	Yes		Not specified

Method	Data Download Functionality	Edge list Construction Functionality	KG Construction Functionality	Constructs Multiple KG Types	Other KG Construction Functionality	Process Ontology Data	Process Linked Open Data	Process Experimental Data	Process Clinical Data	Other Data Types	Data Source Processing Limit
iASiS Open Data Graph	Yes	Yes	Yes	No	Biomedical Harvesters (retrieves articles related to MESH terms, processes OBO ontologies, and pulls data from DrugBank), MedKnow (Uses the UMLS as the backbone and all other data is added as provenance to the edges)	Yes	Yes	No	Yes	Text	Not specified
KG-COVID-19	Yes	Yes	Yes	No	Leverages BioLink	Yes	Yes	No	No		Not specified
Knowledge Base Of Biomedicine (KaBOB)	Yes	Yes	Yes	Yes	Blazegraph	Yes	Yes	No	No		Not specified
Knowledge Graph Exchange (KGX)	Yes	Yes	Yes	Yes	KG verified to confirm it conforms to the Biolink model, KG summary statistics	Yes	Yes	Yes	No		Not specified
Knowledge Graph Toolkit (KGTK)	Yes	Yes	Yes	Yes	Data cleaning module, processes other KGs, KG querying modules, KG. summary statistics, generates node embeddings	No	Yes	No	No	Text	Not specified
PheKnowLator	Yes	Yes	Yes	Yes	Export node metadata Generate property graphs Neo4J Browser	Yes	Yes	Yes	No		Not specified
ProNet	No	Yes	Yes	No	Visualizes a network and enables topological analyses	No	Yes	Yes	No		Not specified
SEmantic Modeling machIne (SEMi)	No	Yes	Yes	Yes	Semantic type detector, multi-edge and weighted graph generator, semantic model builder, link predictor, and semantic model refiner	Yes	Yes	No	No		Not specified

Acronyms: KG: Knowledge Graph.

Note. The following 10 columns were used for scoring the survey: (1) Data downloaded functionality; (2) Edge list construction functionality; (3) KG construction functionality; (4) Constructs multiple KG types; (5) Other KG KG construction functionality; (6) Process ontology data; (7) Process linked open data; (8) Process experiment data; (9) Process clinical data; and (10) Data source processing limit. For scoring, 1 point was awarded for an answer of “Yes” and for the presence of other KG construction functionality.

Supplemental Table 5. Open-Source Survey - Availability.

Method	OpenSource	License Type	Operating Systems	Programming 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	Angulillos, AWS EC2/S3, Titan
BioGrakn	No	None	Linux, Windows, Mac OSX, Cloud-based systems and/or 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
PheKnowLator	Yes	Apache 2.0	Linux, Windows, Mac OSX, Cloud-based systems and/or architectures	Python, Java, Shell	OWL Tools
ProNet	Yes	GPL (>=2)	Linux, Windows, Mac OSX	R	BioGrid, GO

Method	OpenSource	License Type	Operating Systems	Programming Languages	External Dependencies
SEmantic Modeling machIne (SEMi)	Yes	GPL	Linux, Windows, Mac OSX	Python, JavaScript, Shell	Anaconda, Node.js (11.15.0), Java, Maven, Elasticsearch

Note. The following 2 columns were used for scoring the survey: (1) Open Source and (2) License Type. For scoring, 1 point was awarded for an answer of “Yes” and for the presence of a license.

Supplemental Table 6. Open-Source Survey - Usability.

Method	README Provided	Wiki, Read the Docs, or GitPage	Examples of Method Use	Literate Programming Tutorials	Tools to Install Method	Tools to Run Method	Sample Data Provided	Handles Different Sized Data	Output Types	Adoption Indicators
Bio2BEL	Yes	Yes	Yes	No	PyPI Maven	None	Yes	Yes	NetworkX Files, Cytoscape Files, A text file containing edge lists, n-triples files, 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 with preloaded data binaries	Angulillos API Titan	Yes	Yes	Titan	Yes
BioGrakn	Yes	Yes	Yes	Yes	None	Graql Console Grakn Clients Grakn Workbase	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 files, NetworkX Files, a text file containing edge lists, 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

Method	README Provided	Wiki, Read the Docs, or GitPage	Examples of Method Use	Literate Programming Tutorials	Tools to Install Method	Tools to Run Method	Sample Data Provided	Handles Different Sized Data	Output Types	Adoption Indicators
PheKnowLator	Yes	Yes	Yes	Yes	PyPI Jupyter Notebook Docker	Jupyter Notebook Docker	Yes	Yes	RDF/XML, NetworkX, a text file containing edge lists, n-triples, JSON	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

Note. The following 9 columns were used for scoring the survey: (1) README provided; (2) Wiki, Read the Docs, or GitPage; (3) Examples of Method Use; (4) Literate Programming Tutorials; (5) Tools to Install Method; (6) Tools to Run Method; (7) Sample Data Provided; (8) Handles Different Sized Data; and (9) Adoption Indicators. For scoring, 1 point was awarded for an answer of “Yes” and for the presence tools to install and run the method.

Supplemental Table 7. Open-Source 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
PheKnowLator	Yes	1	Yes	Yes	Yes
ProNet	Yes	1	Unclear	No	No
SEmantic Modeling machIne (SEMi)	No	0	Yes	No	No

Note. The following 5 columns were used for scoring the survey: (1) Multiple Releases; (2) Release Count; (3) Method Published; (4) Collaboration Encouraged; and (5) Collaboration Procedures. For scoring, 1 point was awarded for an answer of “Yes” and for having at least one release.

Supplemental Table 8. Open-Source Survey - Reproducibility.

Method	Reproducibility Tools	Installation Tools	Tools to Run Method	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
PheKnowLator	PyPI Docker	Yes	Yes	Yes	Yes	Yes
ProNet	R Markdown	No	No	No	Yes	No
SEmantic Modeling machIne (SEMi)	None	Yes	Yes	No	Yes	Yes

Note. The following 6 columns were used for scoring the survey: (1) Reproducibility Tools; (2) Installation Tools; (3) Tools to Run Methods; (4) Maintainability Measures; (5) Well-Documented Codebase; and (6) Actively Used Issue Tracker. For scoring, 1 point was awarded for an answer of “Yes” and for the presence of at least 1 reproducibility tool.