

# Block 3

## Biological Knowledge Graphs: Examples and Construction Approaches

Tiffany J. Callahan MPH, PhD

Postdoctoral Research Fellow, Department of Biomedical Informatics

*Constructing Knowledge Graphs for Advanced Biomedical Applications*

April 26, 2023

# Lecture Outline

---

## Part 1 - Lecture

- Semantic Web Refresher
- Biological Knowledge Graphs
- Knowledge Graph Evaluation and Open Challenges

## Part 2 - Activity

- Overview

# Part 1 - Lecture



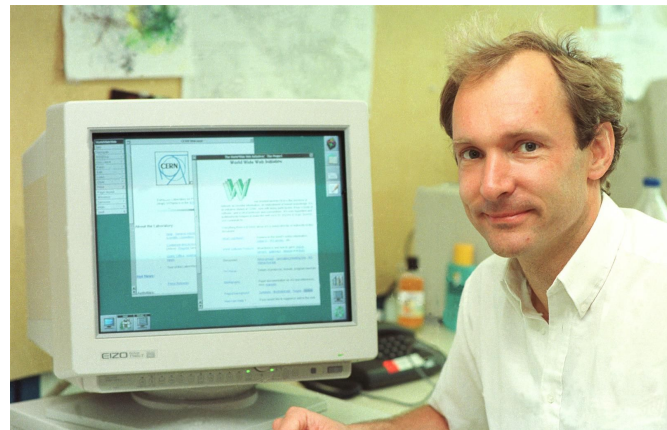
# Semantic Web

---

- The **World Wide Web** (WWW) was invented in 1989 by Tim Berners-Lee to enable rapid sharing of information between scientists across the world
- On the WWW, machines are content brokers storing, organizing, requesting, transmitting, receiving, and displaying content as documents<sup>1</sup>

## Challenges

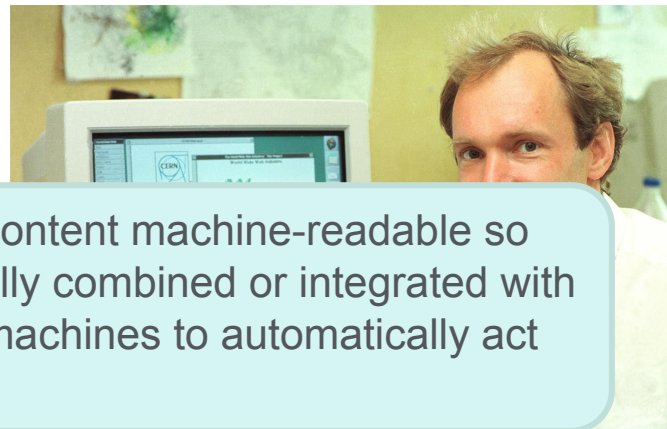
- Content is created for specific websites
- Licensing and technical issues cause fragmentation
- Limited reusability



# Semantic Web

---

- The **World Wide Web** (WWW) was invented in 1989 by Tim Berners-Lee to enable rapid sharing of information between scientists across the world
- On the WWW, machines are content brokers storing,



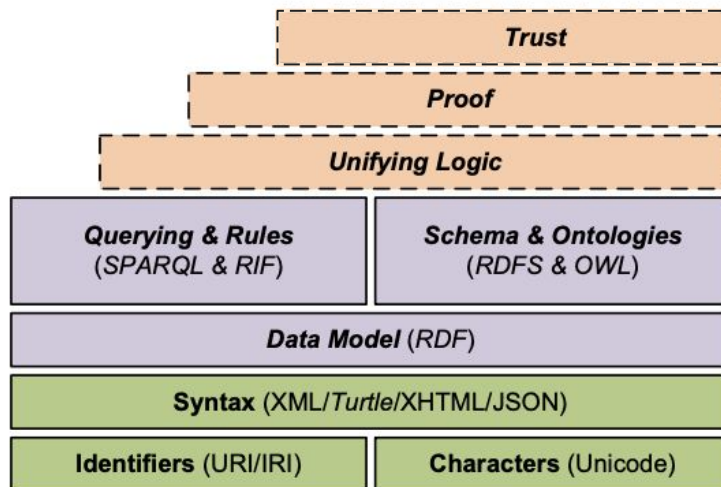
The goal of the **Semantic Web** is to make the Web's content machine-readable so that it can be reused (for any purpose) and automatically combined or integrated with other machine-readable content in order to enable machines to automatically act upon it<sup>1</sup>

- Content is created for specific websites
- Licensing and technical issues cause fragmentation
- Limited reusability

W3C<sup>®</sup>

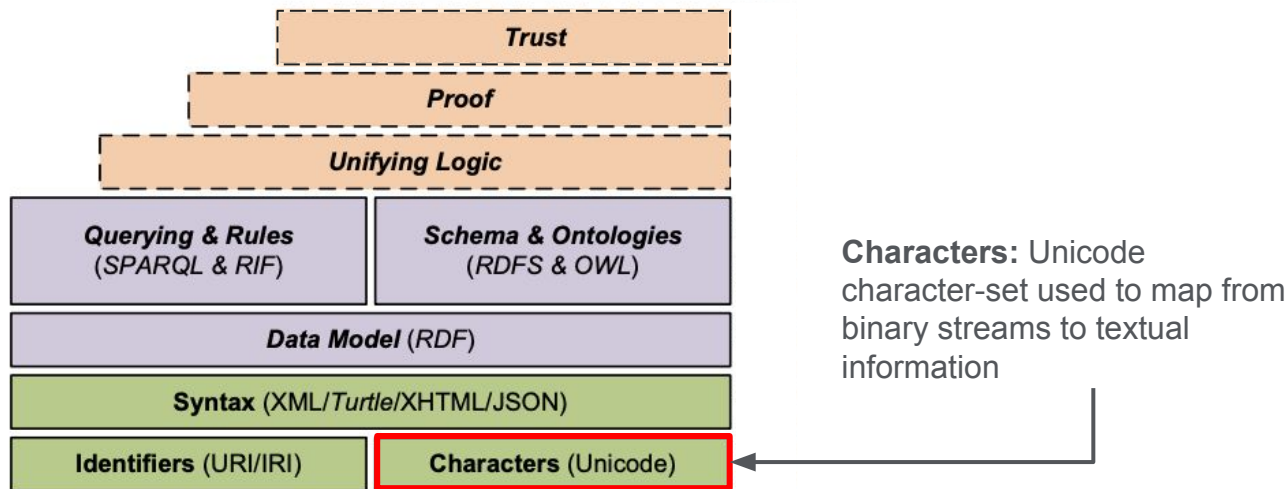
# Semantic Web Standards

---



# Semantic Web Standards

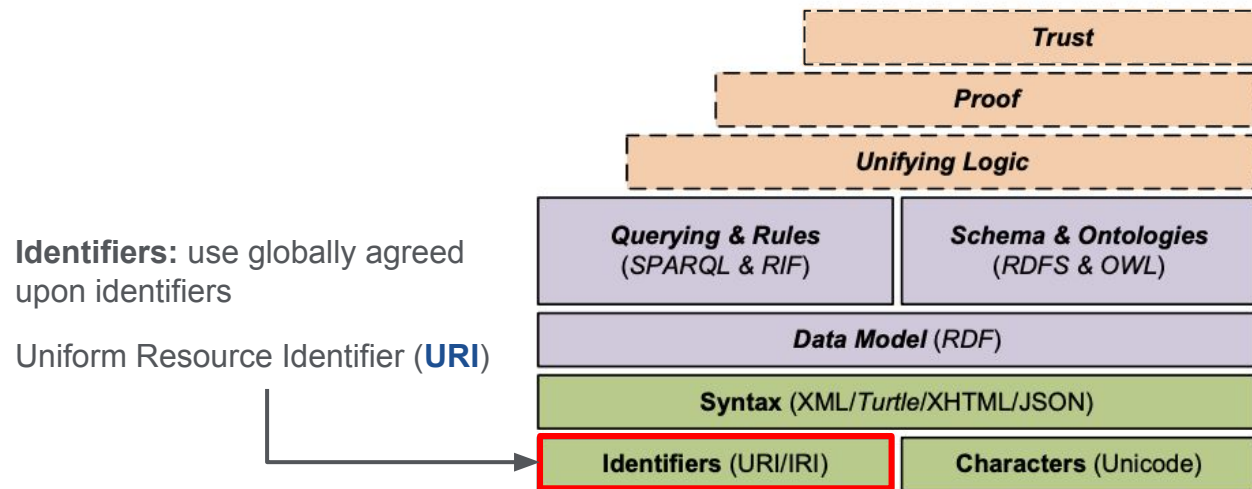
---



Semantic Web Stack  
“Semantic Layer Cake”

# Semantic Web Standards

---

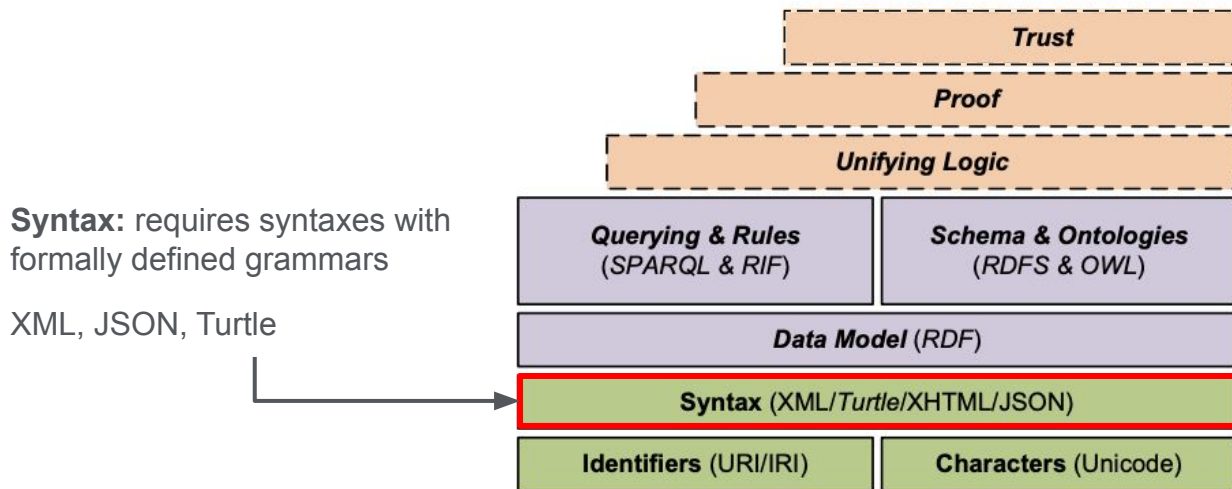


Semantic Web Stack  
"Semantic Layer Cake"



# Semantic Web Standards

---



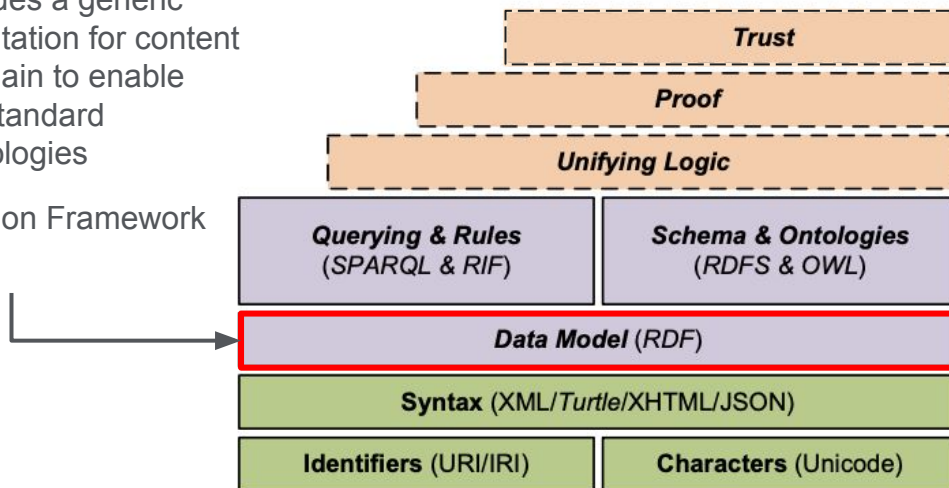
Semantic Web Stack  
"Semantic Layer Cake"

# Semantic Web Standards

---

**Data Model:** provides a generic canonical representation for content irrespective of domain to enable processing using standard off-the-shelf technologies

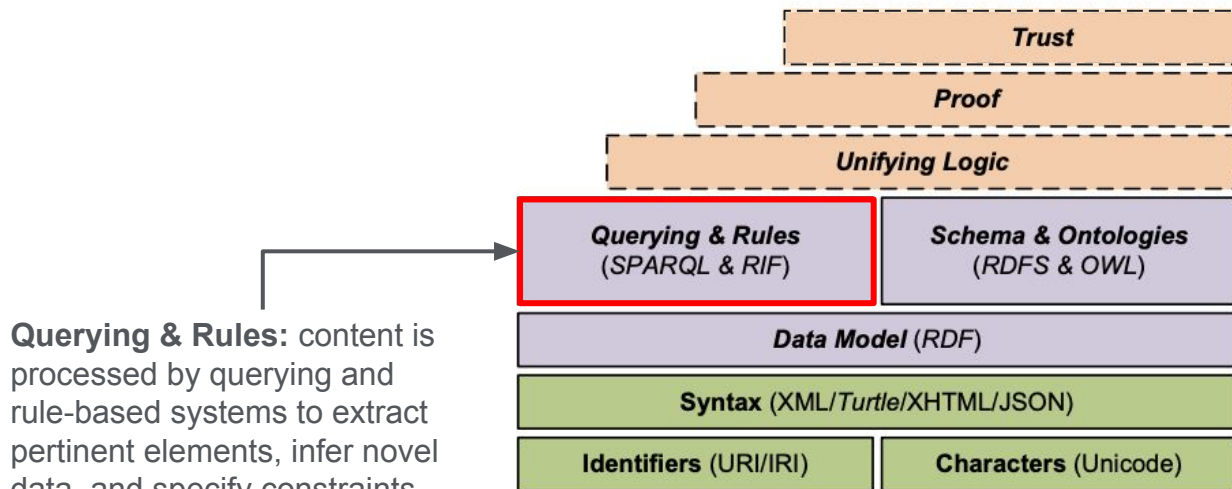
Resource Description Framework  
(**RDF**)



Semantic Web Stack  
“Semantic Layer Cake”

# Semantic Web Standards

---



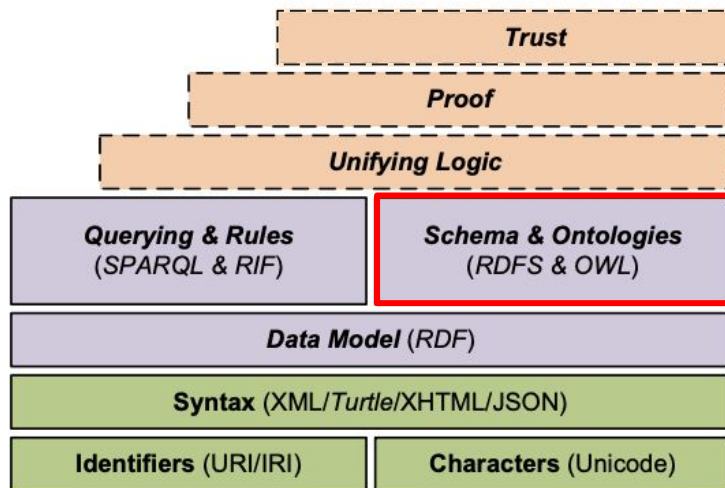
**Querying & Rules:** content is processed by querying and rule-based systems to extract pertinent elements, infer novel data, and specify constraints

SPARQL Protocol and RDF Query Language (**SPARQL**)

Semantic Web Stack  
"Semantic Layer Cake"

# Semantic Web Standards

---



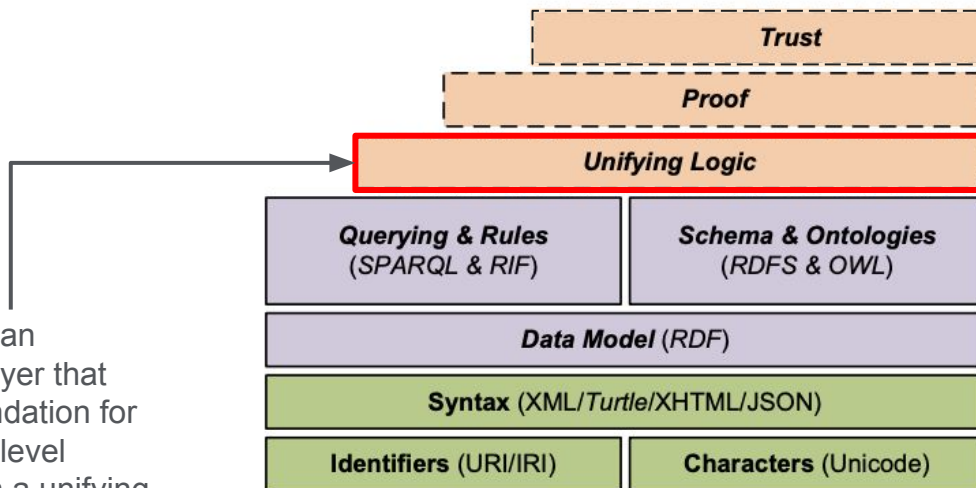
Semantic Web Stack  
"Semantic Layer Cake"

**Schema & Ontologies:** formal languages that provide a meta-vocabulary with well-defined semantics that can be used in combination with the data model

RDF Schema (**RDFS**)  
Web Ontology Language (**OWL**)

# Semantic Web Standards

---



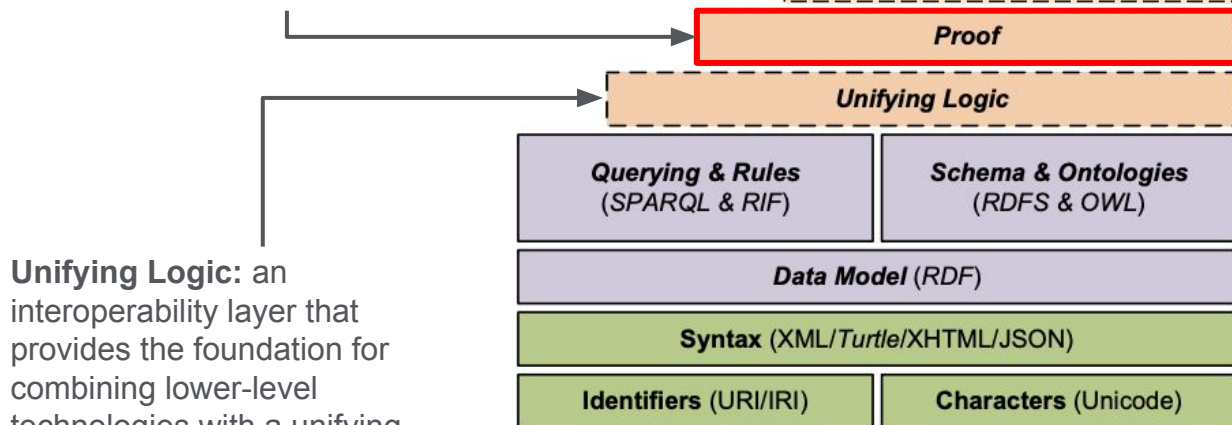
**Unifying Logic:** an interoperability layer that provides the foundation for combining lower-level technologies with a unifying language to engage queries/rules over knowledge represented in RDF and associated ontologies/schemata

Semantic Web Stack  
"Semantic Layer Cake"

# Semantic Web Standards

---

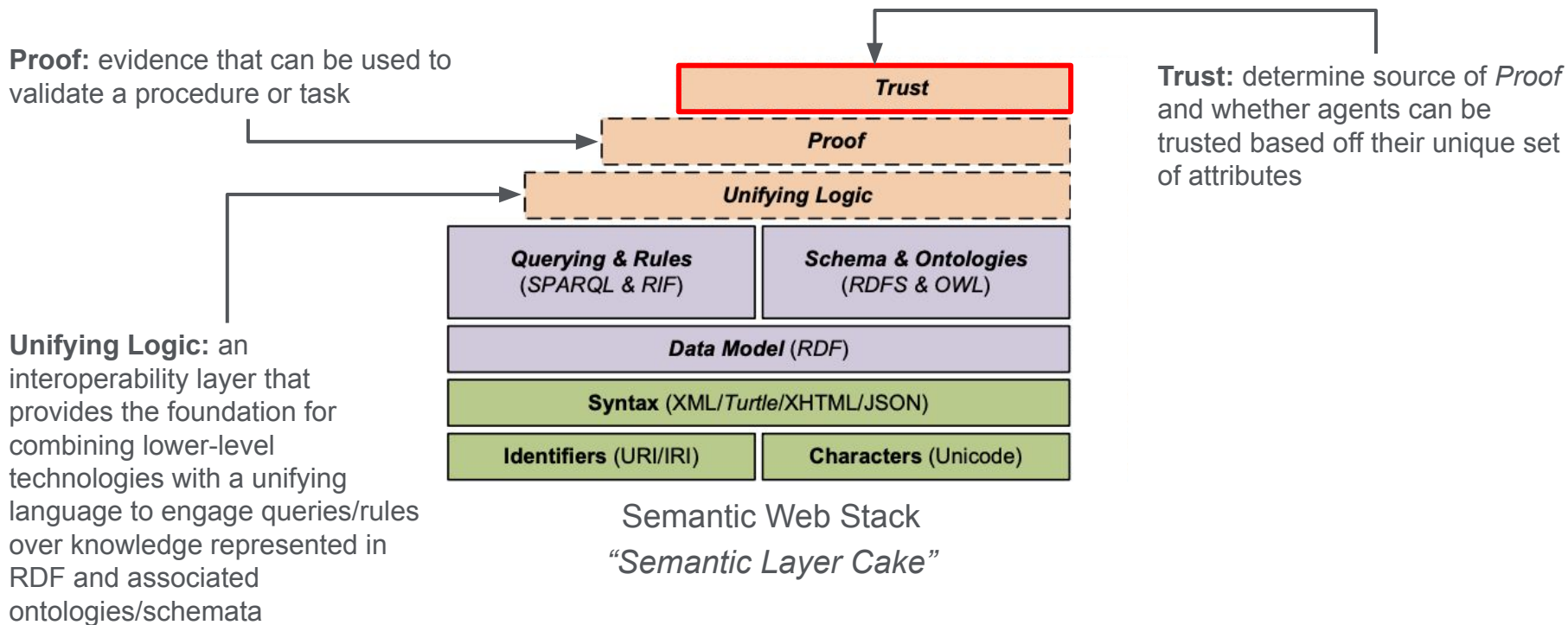
**Proof:** evidence that can be used to validate a procedure or task



**Unifying Logic:** an interoperability layer that provides the foundation for combining lower-level technologies with a unifying language to engage queries/rules over knowledge represented in RDF and associated ontologies/schemata

Semantic Web Stack  
"Semantic Layer Cake"

# Semantic Web Standards



# Semantic Web Standards - RDF

---

The **Resource Description Framework (RDF)** is the core data model for the Semantic Web<sup>1</sup>

## RDF Terms

- **URIs**: universal resource identifiers that identify a resource and its publisher  
[http://purl.obolibrary.org/obo/HP\\_0000726](http://purl.obolibrary.org/obo/HP_0000726) (*dementia*)
- **Literals**: lexical values which can be plain or typed
  - Plain: “dementia” @en
  - Typed: “2” xsd:int (XML Schema definition used to specify variable type)
- **Blank Nodes**: “existential variables” that denote the existence of a resource without having to explicitly reference it with a URI or Literal (*example in 2 slides*)
  - Locally-scoped; usually referenced as a 32-digit hash

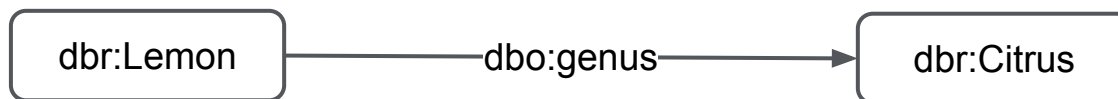


# Semantic Web Standards - Triples

---

RDF **Triples** are used to make statements about “things”

- **Subject:** RDF term (URI or blank node) that refers to the primary resource described by triple
- **Predicate:** RDF term (URI) that identifies the relation between the subject and the object
- **Object:** RDF term (URI, blank node or literal) that fills the value of the relation



# Semantic Web Standards - Blank Nodes

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
ex:LemonadeRecipe ex:steps _:11 .
```

```
_:11 rdf:first ex:SqueezeLemons .
```

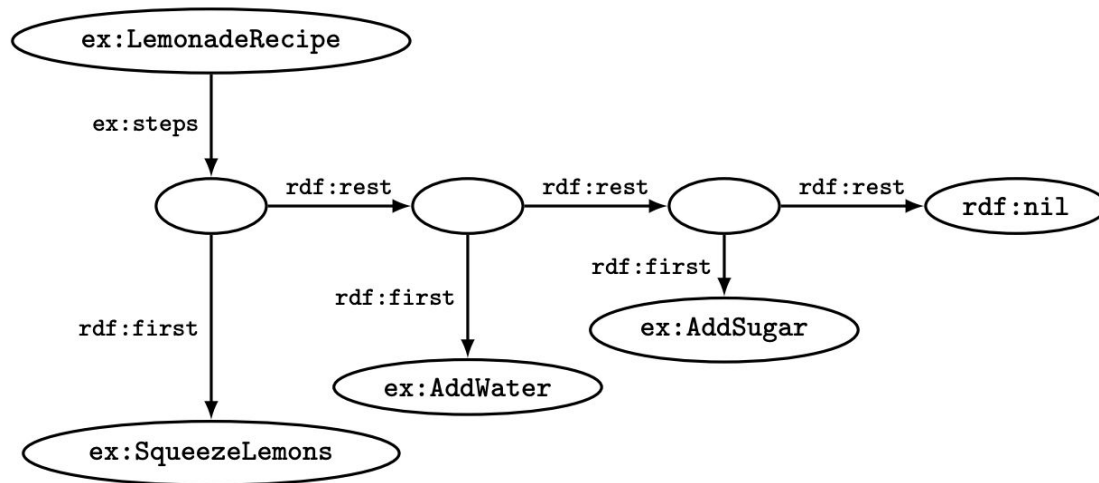
```
_:11 rdf:rest _:12 .
```

```
_:12 rdf:first ex:AddWater .
```

```
_:12 rdf:rest _:13 .
```

```
_:13 rdf:first ex:AddSugar .
```

```
_:13 rdf:rest rdf:nil .
```



# Semantic Web Standards - RDF

---

The **semantics of RDF** are standardized in the form of a model theory

- RDF triples make claims about the nature or configuration of that world<sup>1,2</sup>

**Objective:** formalize claims in a manner that allows for evaluating the: (1) consistency of claims, (3) necessary entailments, and (3) the truth according to the theory

Given an RDF graph containing claims held as true, **entailments** are logical deductions implied by these claims

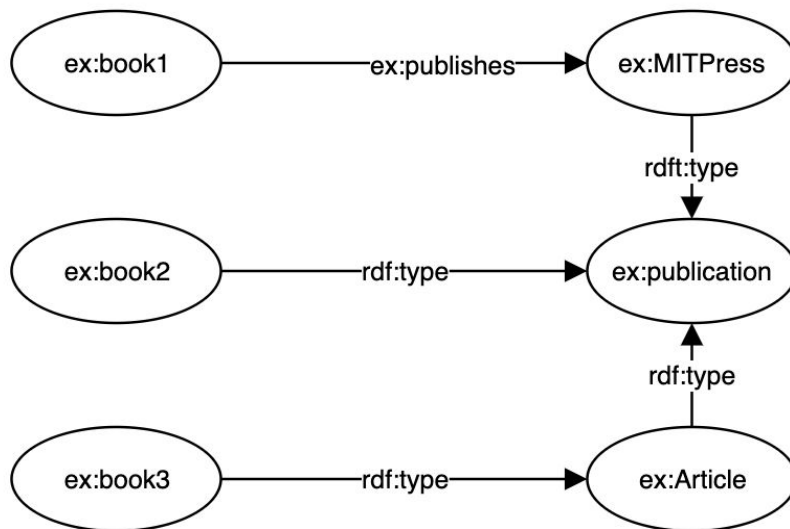
- Foundation for machine-readability
- Lets machines “connect the dots”

**Example:**

- “p53 is a tumor suppressor gene” entails “p53 is a gene” since p53 cannot be a tumor suppressing gene without also being a gene

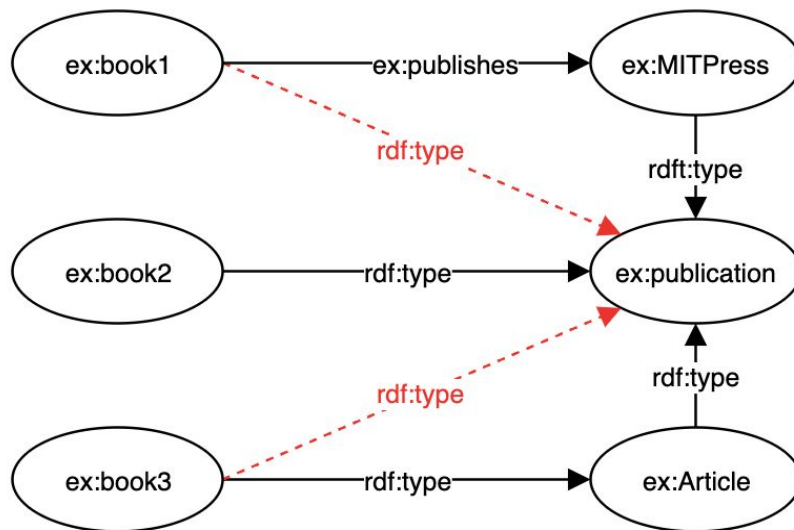
# Semantic Web Standards - Entailment

---



# Semantic Web Standards - Entailment

---



# Semantic Web Standards - RDFS

---

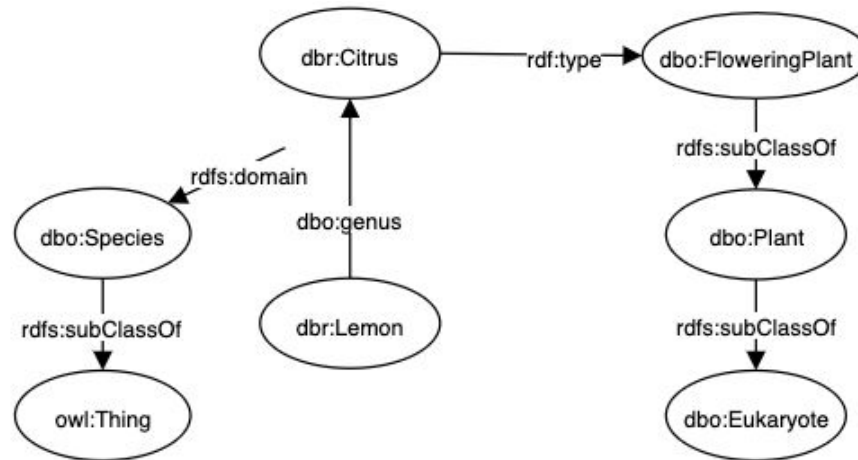
The **RDF Schema (RDFS)** attaches semantics to the RDF vocabulary.<sup>1,2</sup>

Extends the RDF Vocabulary in four key ways:

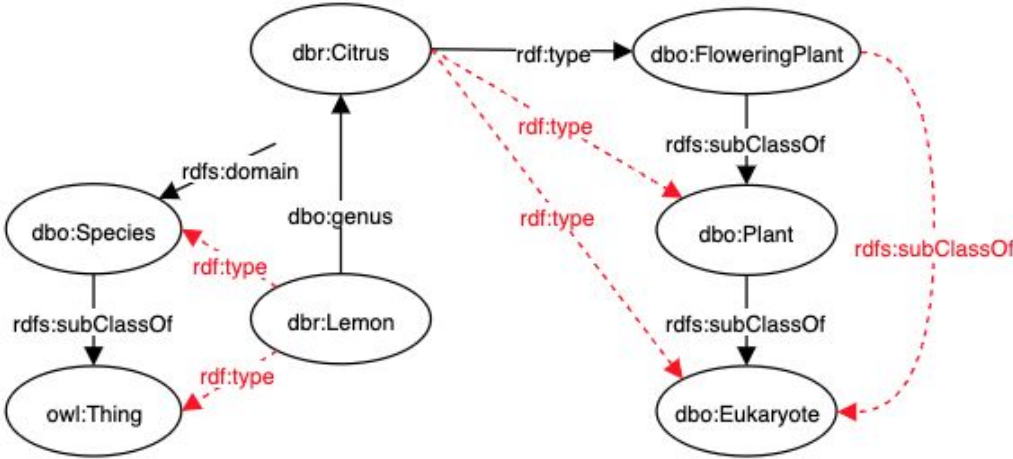
- **rdfs:subClassOf**: allows for stating that the extension of one class *c1* (its set of members) is necessarily contained within the extension of another class *c2*
- **rdfs:subPropertyOf**: allows for stating that all things related by a given property *p1* are also necessarily related by another property *p2*
- **rdfs:domain**: allows for stating that the subject of a relation with a given property *p* is a member of a given class *c*
- **rdfs:range**: analogously allows for stating that the object of a relation with a given property *p* is a member of a given class *c*

# Semantic Web Standards - Entailment

---



# Semantic Web Standards - Entailment





# Semantic Web Standards - OWL

---

The **Web Ontology Language (OWL)** enables more expressive semantics and richer entailment<sup>1,2</sup>

Extends the RDFS with things like:

- **EquivalentClass/Property**: allows for stating two classes/properties are the same
- **disjointWith/disjointPropertyWith**: allows for stating two classes/properties never the same
- **TransitiveProperty**: enables inference based on transitive law
- **UnionOf/intersectionOf/ComplementOf**: enables the application of set logic

# Semantic Web Standards - OWL

---

**OWL** supports “RDF-based Semantics”,

Supports “direct semantics” enabling ontologies to be translated into **axioms** compatible with a formalism called Description Logics<sup>1</sup>

**Description Logics** aim to define a subset of First Order Logic where the semantics of the language can be supported in a sound (correct) and complete manner using algorithms or reasoners<sup>2</sup>

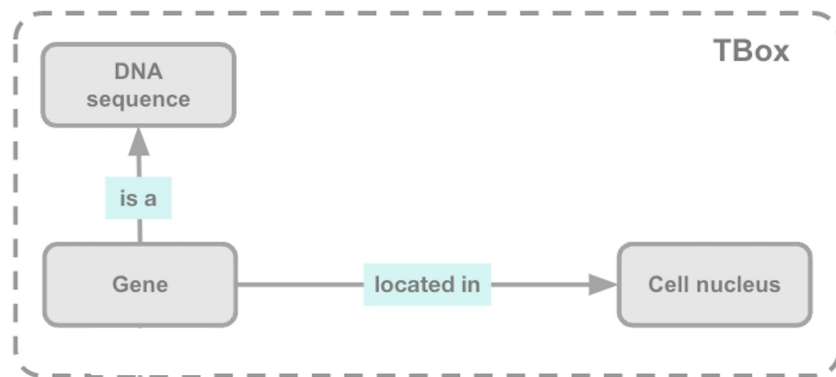
Reasoners can verify:

consistency, satisfiability (class membership does not cause logical contradictions), subsumption (a class is necessarily a subclass of another), instance (a resource is a member of a class) and conjunctive query answering (complex queries against the ontology and its entailments)<sup>3</sup>

# Semantic Web Standards - Description Logics

**Description logics** splits concepts and their relationships from instances and their attributes/roles

**TBox** describes “classes” (i.e., types of objects), properties (i.e., relationships), and assertions (i.e., statements of facts) assumed to generally be true. The TBox is often formalized as an ontology

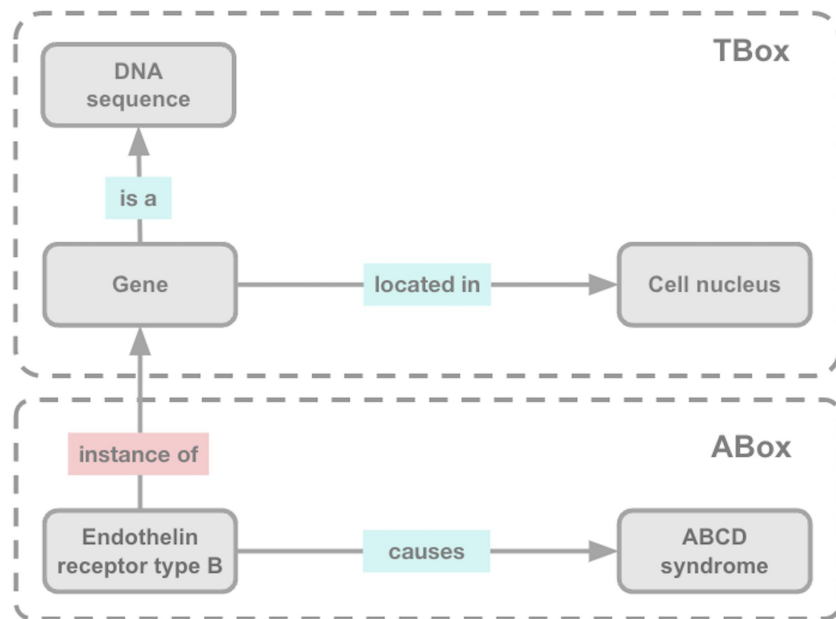


# Semantic Web Standards - Description Logics

**Description logics** splits concepts and their relationships from instances and their attributes/roles

**TBox** describes “classes” (i.e., types of objects), properties (i.e., relationships), and assertions (i.e., statements of facts) assumed to generally be true. The TBox is often formalized as an ontology

**ABox** describes “individuals” or instances of classes and assertions that are specific to an instance. The ABox models instances of ontology classes and properties



## Biological Knowledge Graphs



# Knowledge-based Biomedical Data Science

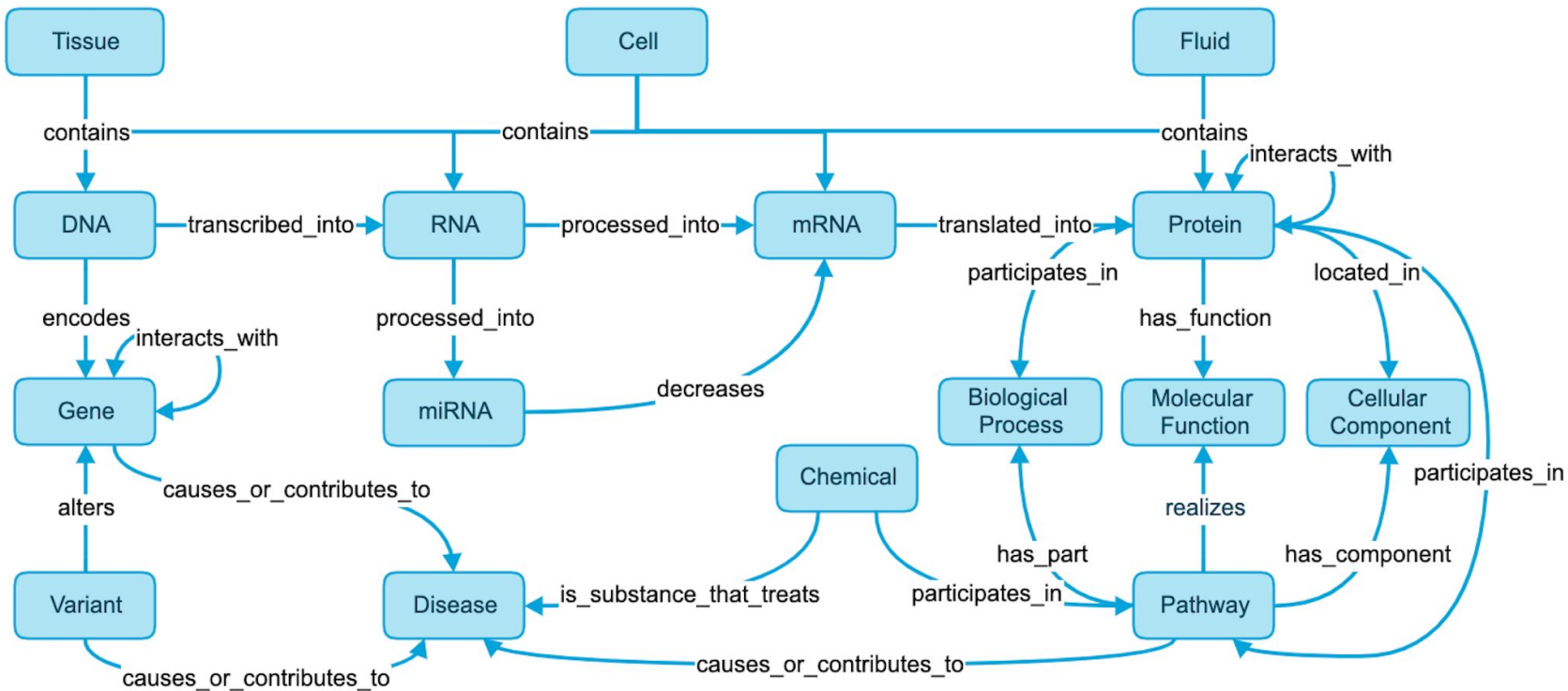
---

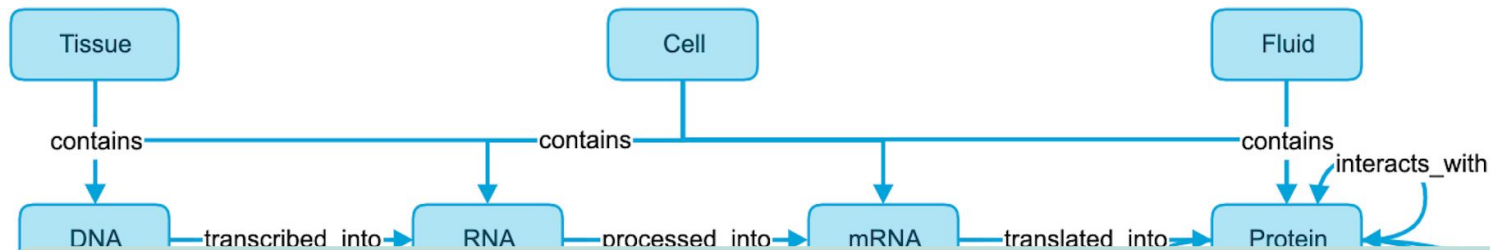
**Knowledge-based biomedical data science** involves designing computer systems to act as if they know about medicine and biology<sup>1</sup>

There are many ways in which a system might act as if it knows something:

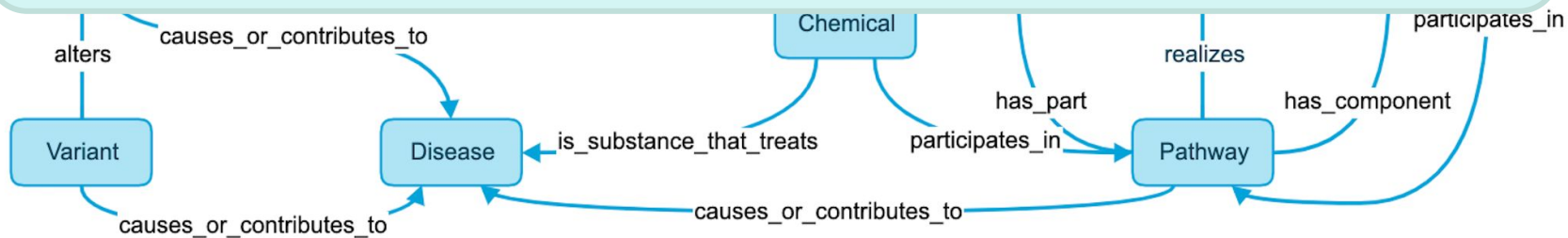
- Use existing knowledge to generate, rank, or evaluate hypotheses about a dataset
- Answer a natural language question about a biomedical topic

Knowledge-based systems specify a **knowledge representation**—how a computer system represents knowledge internally—and one or more **inference or reasoning methods**—how computations over knowledge representations are used to produce output<sup>2</sup>





How are knowledge graphs typically used in the biomedical domain?



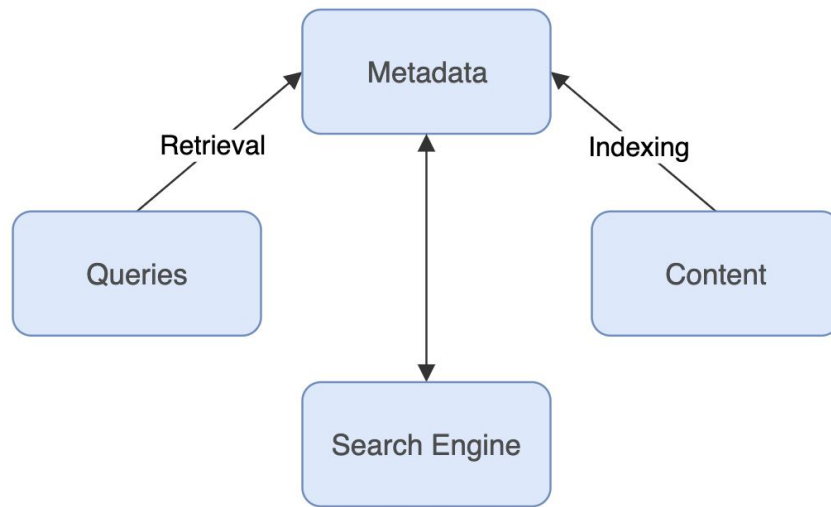


# Biomedical Knowledge Graphs - Applications

---

## Information Retrieval

- Knowledge graphs have been used to organize knowledge for information retrieval
- Designed to make it possible to find facts or evidence for a wide variety of topics



# Biomedical Knowledge Graphs - Applications

---

## Inferring New Knowledge

- Graph Algorithms
  - Edge (or link) prediction and community detection or clustering. These methods are a form of hypothesis generation and often include an estimate of confidence in the prediction

# Biomedical Knowledge Graphs - Applications

---

## Inferring New Knowledge

### - Graph Algorithms

- Edge (or link) prediction and community detection or clustering. These methods are a form of hypothesis generation and often include an estimate of confidence in the prediction

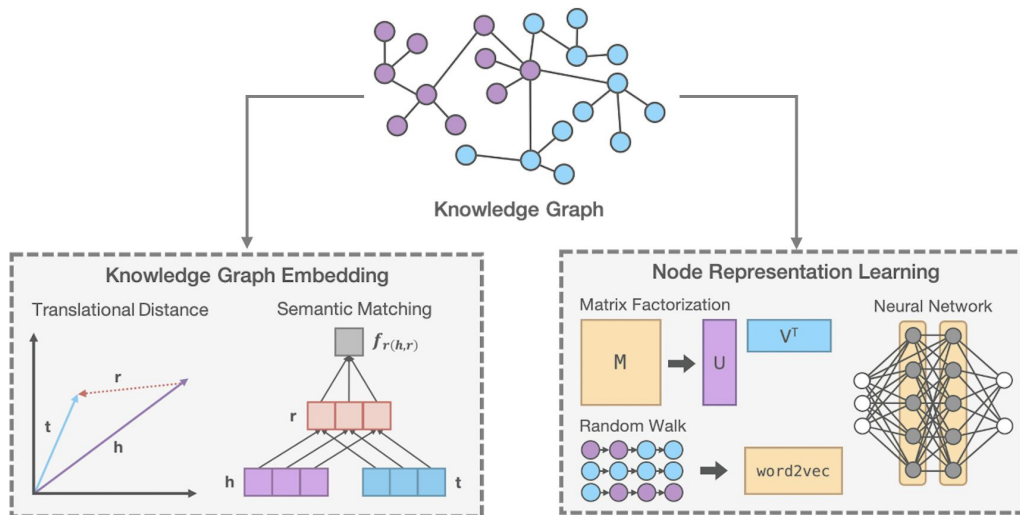
### - Logical Reasoners

- Satisfiability inference checks to see if a class definition is logically satisfiable
- Subsumption inference uses class definitions to identify all classes that are fully contained within some other class. Particularly useful because it makes explicit many edges that are otherwise implicit, and therefore it can improve the results of other algorithms that depend on the structure of the graph, such as link prediction or embeddings

# Biomedical Knowledge Graphs - Applications

## Alternative Representations

- Neural networks are commonly applied to knowledge graphs to create embeddings which can be used as input for downstream learning and Q&A systems



# Biomedical Knowledge Graphs - Clinical Applications

---

## Clinical

Using knowledge graphs with traditional rule-based approaches for information retrieval

- Graph-based patient representations to enable querying based on domain knowledge<sup>1</sup>
- Evidence for diagnostic assistance, clinical decision support machinery, or surveillance
  - Predict treatments for and causes of different diseases<sup>2,3</sup>
  - Pharmacovigilance and drug safety surveillance systems<sup>4,5</sup>

Discover missing knowledge or generate novel hypotheses

- Identify comorbid diseases<sup>6,7</sup> and drug repurposing candidates<sup>8</sup>

Capture complex patient information for further processing

- Create patient-specific KGs for analysis and visualization<sup>9-11</sup>
- Interactive tool to help users interact with complex data<sup>12</sup>

# Biomedical Knowledge Graphs - Biological Applications

---

## Biological

Applying node embedding techniques to knowledge graphs

- For prediction or visualization of complex biological systems in low-dimensional spaces<sup>15,45,46</sup>
- To convert knowledge graphs into low-dimensional spaces in order to visualize clusters in two- or three-dimensional projections to better display entities of interest<sup>41</sup>
- To leverage semantic similarity-inspired hypotheses and identify valuable drug–drug<sup>40,47,51</sup>, drug–target<sup>51</sup>, or protein–protein interactions<sup>46,48</sup>
- To perform link prediction methods in order to test hypothesize previously hypothesized and/or unobserved biological relationships for drug repurposing<sup>36,40,45,47-56</sup>
- To combine with gene expression time series data in order to create specific and detailed hypotheses regarding mechanisms of toxicity<sup>45</sup>

# Biomedical Knowledge Graphs - NLP

---

## Natural Language Processing (NLP)

Knowledge graphs have been used to improve NLP performance

- Summarization or information extraction from EHRs and Q&A systems<sup>15,26,27,29,40,60,61</sup>
- Knowledge graph-derived embeddings used alone or in combination with other text-derived features<sup>46</sup> improved the performance of a variety of NLP tasks, including named entity recognition,<sup>62</sup> coreference resolution,<sup>63</sup> and relation extraction<sup>64</sup>

Knowledge graphs are important components of information extraction systems

- Ontologies can serve as formal dictionaries allowing for rapid indexing in named entity recognition and word sense disambiguation tasks<sup>65, 66</sup>
- Knowledge graphs offer richer semantic context than lexicons, identifying not only similar concepts but also rich collections of relationships that can be used to disambiguate or otherwise improve concept recognition in texts<sup>65-68</sup>

# Biomedical Knowledge Graphs - Organizational Efforts

---

The US and European scientific institutions support knowledge graph efforts

- The National Institutes of Health's National Center for Advancing Translational Sciences' Biomedical Data Translator project (<https://ncats.nih.gov/translator>)
  - A computational system that integrates sources of existing biomedical knowledge in order to translate clinical inquiries into relevant research results that synthesize elements of the integrated knowledge to directly answer the inquiry or generate testable hypotheses
- Elixir Europe (<https://elixir-europe.org/>)
  - Managing and safeguarding data generated by publicly funded life science research and integrating bioinformatics resources. The Elixir Core Data Resources are leaders in the production of interoperable knowledge resources and are widely used components of biomedical knowledge graphs



## Biological Knowledge Graphs



# Constructing Knowledge Graphs

---

Knowledge graphs can be built in many ways. There is no single best representation that will solve all graph-based problems in the Life Sciences; different representations are needed to solve different problems.

## Approaches

- Simple (Property Graphs): joining nodes as a series of edges
- Hybrid RDF Graphs: joining nodes as a series of edges to existing ontologies
- Complex (OWL Graphs): use of formal semantics with or without existing ontologies

## General Build Workflow

- Download data
- Preprocess data (filtering, identifier mapping and cross-walking)
- Construct graph
- Prepare/incorporate node/edge metadata (if included)
- Serialize output

Legend

Cross Domain

Geography

Government

Life Sciences

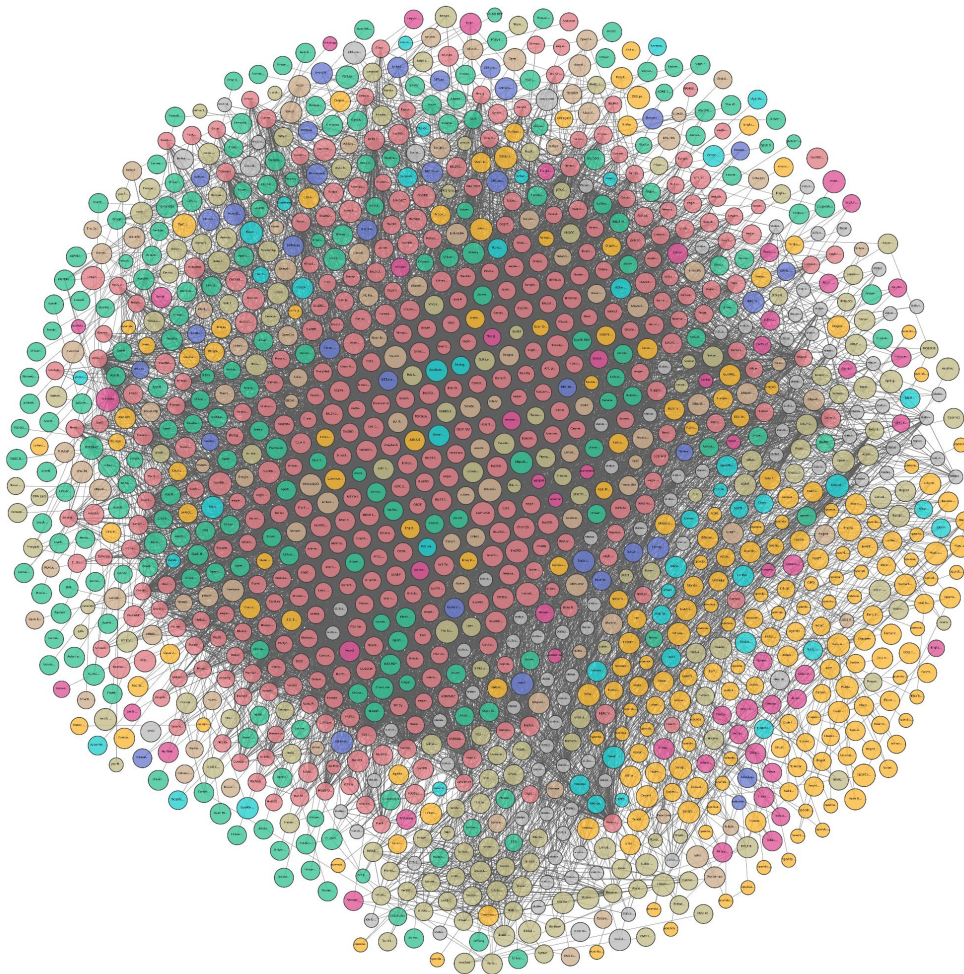
Linguistics

Media

Publications

Social Networking

User Generated



# Constructing Knowledge Graphs- Linked Open Data

---

The core aim of **Linked Data** is to provide a set of principles by which the Semantic Web standards can be effectively deployed on the Web to facilitate discovery and interoperability of structured data<sup>1</sup>

Early RDF data were dumped into “silos”, which while the data within them were interoperable they were rarely interlinked, had inconsistent URI naming, and were often published using different conventions<sup>1</sup>

**Linked Data Principles** were created in 2006 to help address these limitations:

- Open Data “5 Star Scheme”, where each star increases the **reusability and interoperability**<sup>1,2</sup>

- 
- ★ PUBLISH DATA ON THE WEB UNDER AN OPEN LICENSE
  - ★★ PUBLISH STRUCTURED DATA
  - ★★★ USE NON-PROPRIETARY FORMATS
  - ★★★★ USE URIs TO IDENTIFY THINGS
  - ★★★★★ LINK YOUR DATA TO OTHER DATA
-

# Constructing Knowledge Graphs - Important Assumptions

---

## Open World Assumption

Given a knowledge base, a statement can be **True**, **False**, or **Unknown**

**True**: a statement can always be derived from the knowledge base

**False**: a statement can never be derived from the knowledge base

**Unknown**: a statement can be derived by a version of the knowledge base

**Assumption:** knowledge base only covers key aspects of world

## Closed World Assumption

Given a knowledge base, a statement can only be **True** or **False**

**True**: a statement can always be derived from the knowledge base

**False**: otherwise

**Assumption:** knowledge base has complete knowledge about part of the world

# Constructing Knowledge Graphs - Important Assumptions



## Open World Assumption

Given a knowledge base, a statement can be **True**, **False**, or **Unknown**

**True**: a statement can always be derived from the knowledge base

**False**: a statement can never be derived from the knowledge base

**Unknown**: a statement can be derived by a version of the knowledge base

**Assumption:** knowledge base only covers key aspects of world

## Closed World Assumption

Given a knowledge base, a statement can only be **True** or **False**

**True**: a statement can always be derived from the knowledge base

**False**: otherwise

**Assumption:** knowledge base has complete knowledge about part of the world

# Constructing Knowledge Graphs - Simple

---

**Simple knowledge graphs** aim to model entities and their relations using (basic) network science representations without formal semantics

- One or more node types
- A single directed or undirected edge
- Associated node and edge metadata

**Build Workflow:** General

**Output:** flat-file (tabular) edge list and node data; adjacency matrix; metadata annotations

# Constructing Knowledge Graphs - Simple



Edges

source	target
9021	GO:0003193
UBERON:0000004	3149
UBERON:0001891	D058447
2063	7846

Nodes

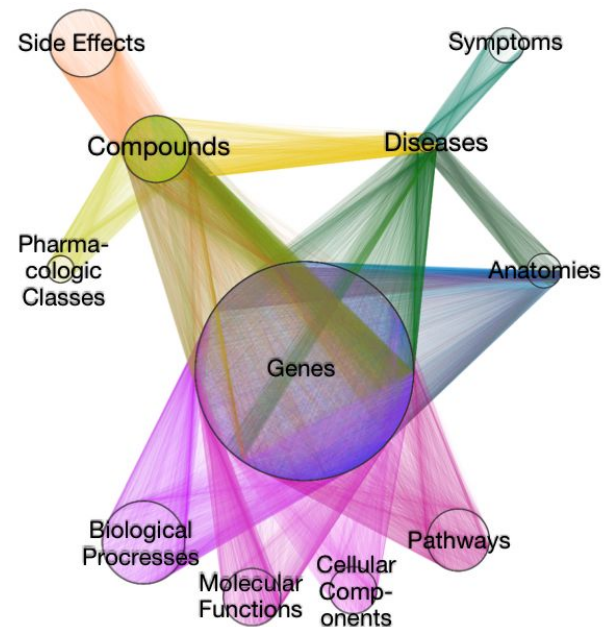
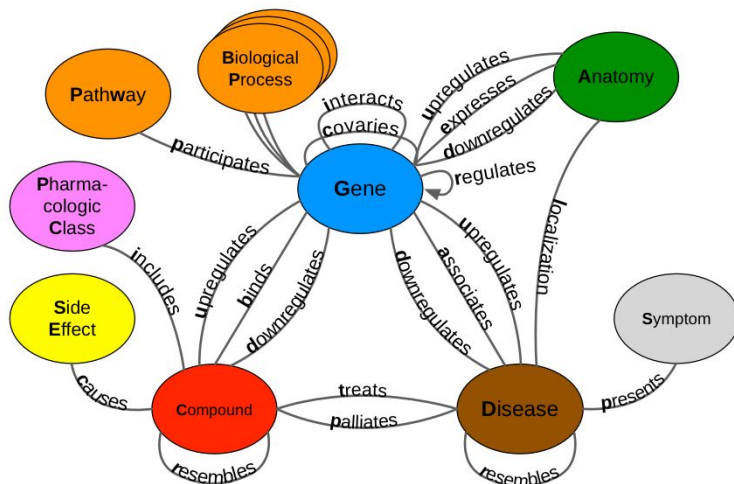
id	name	kind
UBERON:0000004	nose	Anatomy
GO:0003193	pulmonary valve formation	Biological Process
3149	HMGB3	Gene
D058447	Eye Pain	Symptom



# Hetionet

**Hetionet** is a heterogeneous network with multiple node and edge (relationship) types. The hetnet was designed for **Project Repheto**, which aims to systematically identify why drugs work and predict new therapies for drugs.

47,031 nodes (11 types) and 2,250,197 relationships (24 types)



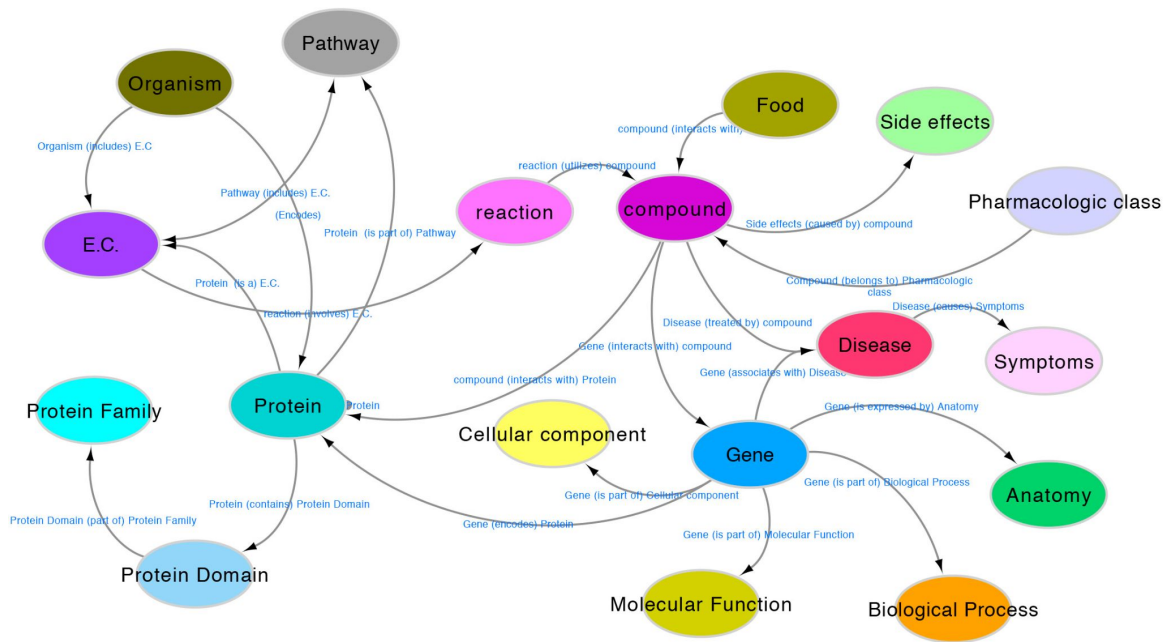
<https://het.io/>

# Scalable Precision Medicine Open Knowledge Engine

## Scalable Precision Medicine Open Knowledge Engine (SPOKE)

currently includes 19 databases such as LINCS, GWAS Catalog, ChemBL, DrugBank, SIDER and iRefIndex. The clinical insight sources include UCSF's large data store of de-identified patient clinical data. Designed to connect basic molecular data with clinical insights and environmental exposures.

47,000 nodes (11 types) and 2.25 million edges (24 types)

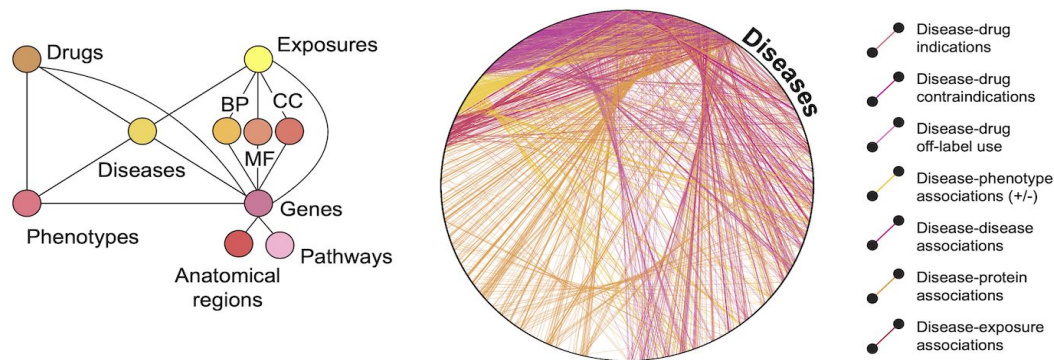


<https://spoke.ucsf.edu/>

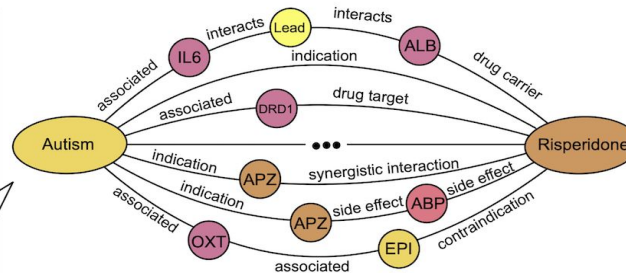
# PrimeKG

PrimeKG is a “a precision medicine-oriented knowledge graph that provides a holistic view of diseases”. The graph integrates 20 resources to describe diseases with relationships representing 10 major biological scales and all approved and experimental drugs.

129,375 nodes (10 types) and 8,100,498 edges (30 types)



A spectrum of developmental disorders that includes autism and Asperger syndrome. Signs and symptoms include poor communication skills, defective social interactions, and repetitive behaviors. Each child with autism spectrum disorder is likely to have a unique pattern of behavior [...] Autism spectrum disorder has no single known cause. [...] Autism spectrum disorder affects children of all races and nationalities, but certain factors increase a child's risk [...] There's no way to prevent autism spectrum disorder, but there are treatment options.



Risperidone is a second-generation antipsychotic (SGA) medication used in the treatment of a number of mood and mental health conditions including schizophrenia and bipolar disorder. The half-life is 3 hours in extensive metabolizers. Though its precise mechanism of action is not fully understood, current focus is on the ability of risperidone to inhibit the D2 dopaminergic receptors and 5-HT2A serotonergic receptors in the brain. [...] Risperidone and its active metabolite, 9-hydroxyrisperidone, are ~88% and ~77% protein-bound in human plasma, respectively. [...] The primary action of risperidone is to decrease dopaminergic and serotonergic pathway activity in the brain, therefore decreasing symptoms of schizophrenia and mood disorders.

<https://zitniklab.hms.harvard.edu/projects/PrimeKG/>

# Constructing Knowledge Graphs - Hybrid

---

**Hybrid knowledge graphs** aim to model entities and their relations using a mix of standard network representations and “some” formal semantics (usually RDF) borrowed from ontologies

- One or more node types (typed)
- A single directed or undirected edge (typed)
- Associated node and edge metadata

**Build Workflow:** General

**Output:** flat-file (tabular) edge list and node data; adjacency matrix; metadata annotations

# Constructing Knowledge Graphs - Hybrid

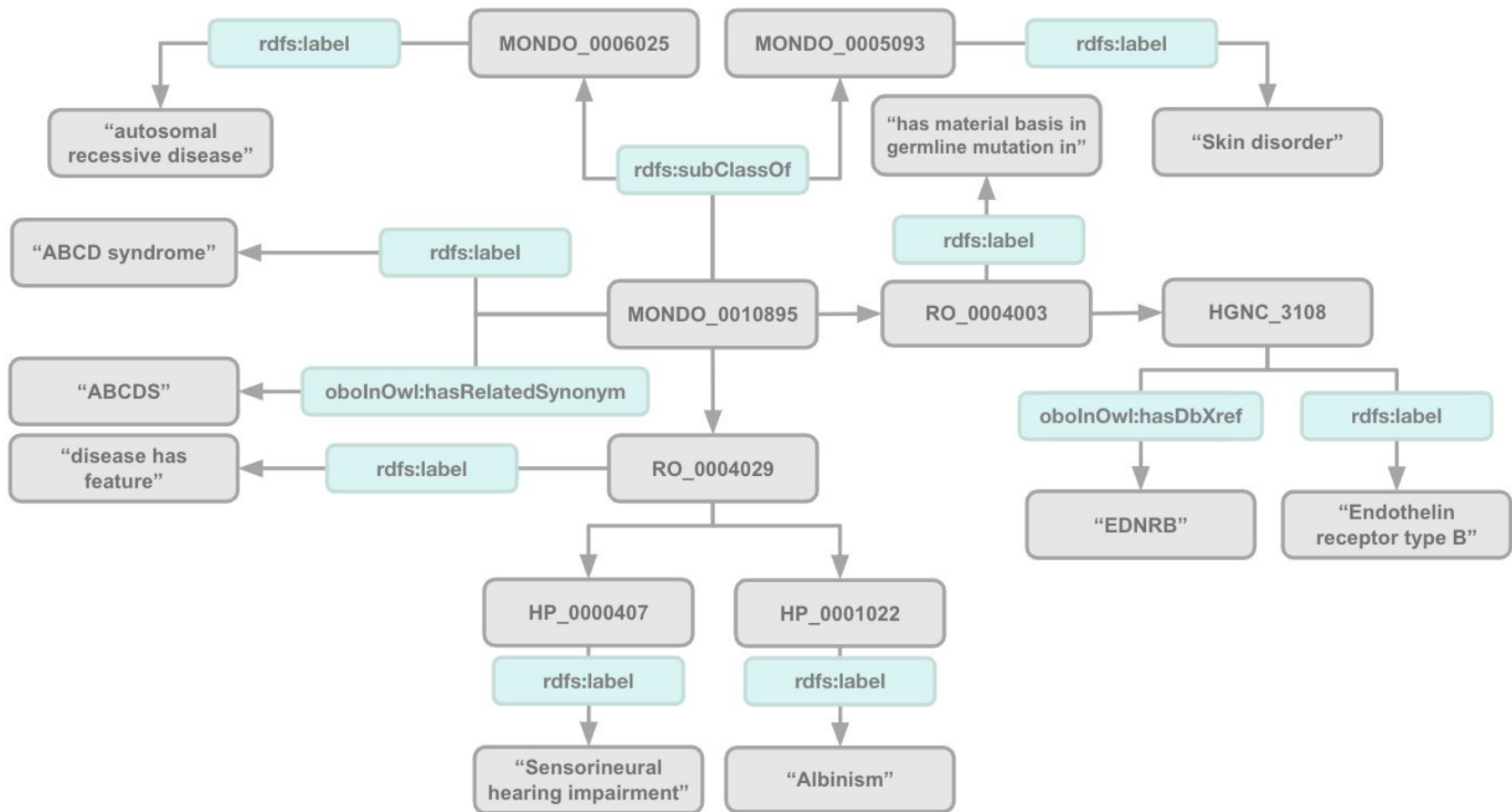
---

Edges

source	metaedge	target
9021	GpBP	GO:0071357
UBERON:0002081	AeG	8519
UBERON:0001891	AeG	84281
2063	Gr>G	Gene:7846

Nodes

id	label	type
UBERON:0000004	nose	Anatomy
GO:0003193	pulmonary valve formation	Biological Process
3149	HMGB3	Gene
D058447	Eye Pain	Symptom

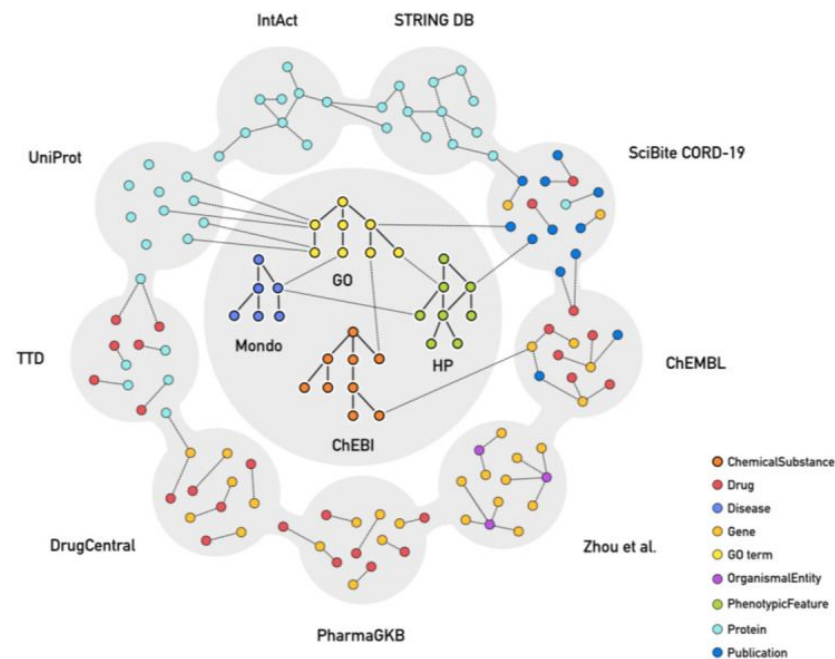




# KG-COVID-19

**KG-COVID-19** is a framework developed and maintained by the **Monarch Initiative** that ingests and integrates different sources of biomedical data to produce knowledge graphs for COVID-19 response. “This framework can also be applied to other problems in which siloed biomedical data must be quickly integrated for different research applications, including future pandemics.”

574,778 nodes (20 types) and 24,145,561 edges (561 types)



<https://github.com/Knowledge-Graph-Hub/kg-covid-19/wiki>



# Constructing Knowledge Graphs - Complex

---

**Complex knowledge graphs** aim to model entities and their relations using formal semantics with explicit modifications to data not represented as an ontology

- One or more node types (RDF typed)
- Multiple directed edges (RDF typed)
- Associated node and edge metadata (integrated as annotation assertions or stored as named graphs)

**Build Workflow:** General + modifications to “ontologize” data not represented as an ontology

**Output:** Turtle, RDF/XML, OWL

# Constructing Knowledge Graphs - Complex

---

Nodes

type	id	label	rdf:type
GO:0003193	pulmonary valve formation	Biological Process	
Node	ncbi_gene:7040	TGFB1	gene
D058447	reactome:R-HSA-168277	Influenza Virus Induced Apoptosis	pathway
Node	obo:PW_0001054	Influenza A Pathway	pathway
Relation	obo:RO_0000056	participates in	gene-pathway

Edges

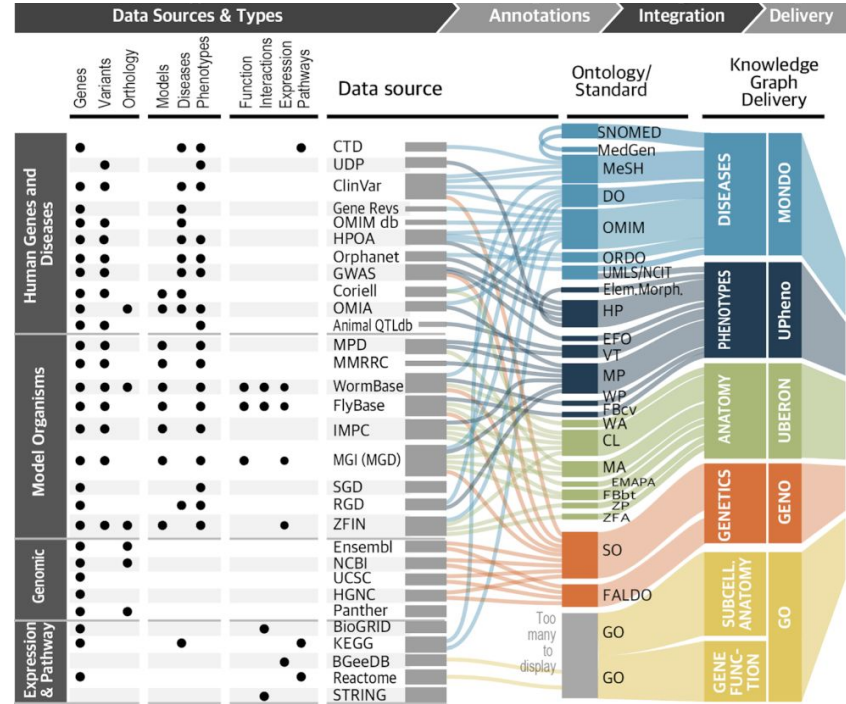
subject	predicate	object
reactome:R-HSA-168277	rdfs:subClassOf	obo:PW_0001054
reactome:R-HSA-168277	rdf:type	owl:Class
BlankNode1	rdfs:subClassOf	ncbi_gene:7040
BlankNode1	rdfs:subClassOf	BlankNode2
BlankNode2	rdf:type	owl:Restriction
BlankNode2	owl:someValuesFrom	reactome:R-HSA-168277
BlankNode2	owl:onProperty	obo:RO_0000056



# Monarch Knowledge Graph

The **Monarch Knowledge Graph** is part of a large system designed and maintained by the **Monarch Initiative** with the goal of representing gene-to-phenotype relationships across multiple model organisms. The Monarch platform is used to construct this knowledge graph, which are used to construct several different ontologies, which can also be integrated with each other.

32.9 million nodes and 160,000,000 million edges

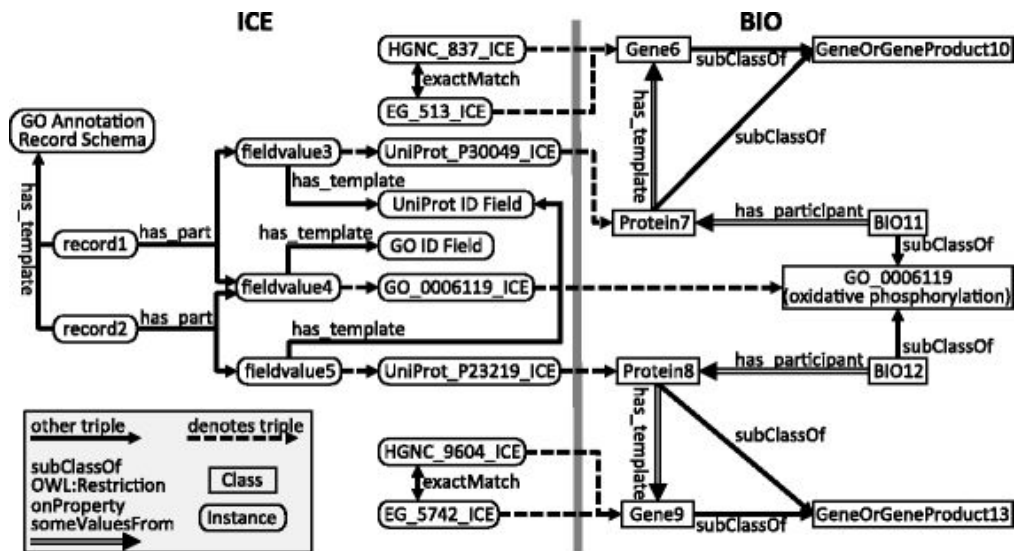


<https://archive.monarchinitiative.org/latest>

# KaBOB

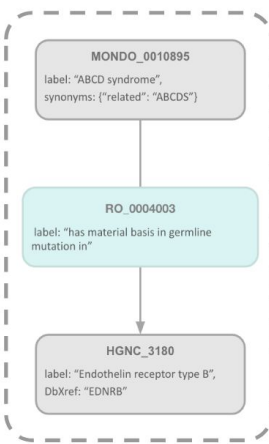
The **Knowledge Base of Biomedicine (KaBOB)** is a large-scale semantically rich representation of important biomedical concepts and their relationships. KaBOB is grounded in the Open Biomedical and Biological Ontology Foundry Ontologies and represents data for seven model organisms.

~500 million triples (according to 2015 publication)

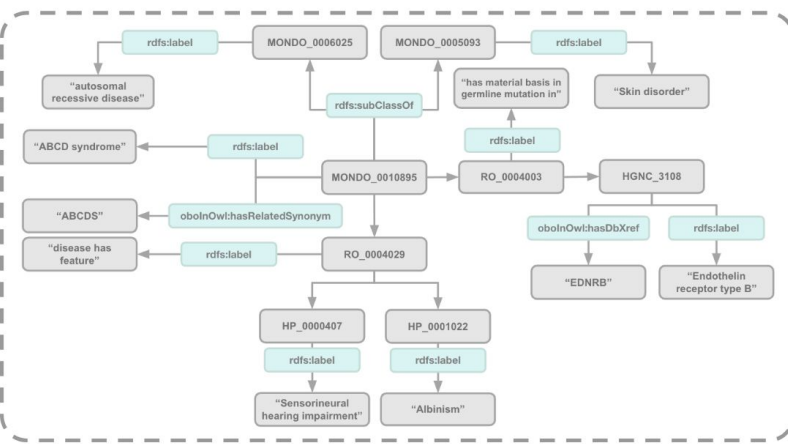


<https://github.com/drlivingston/kabob>

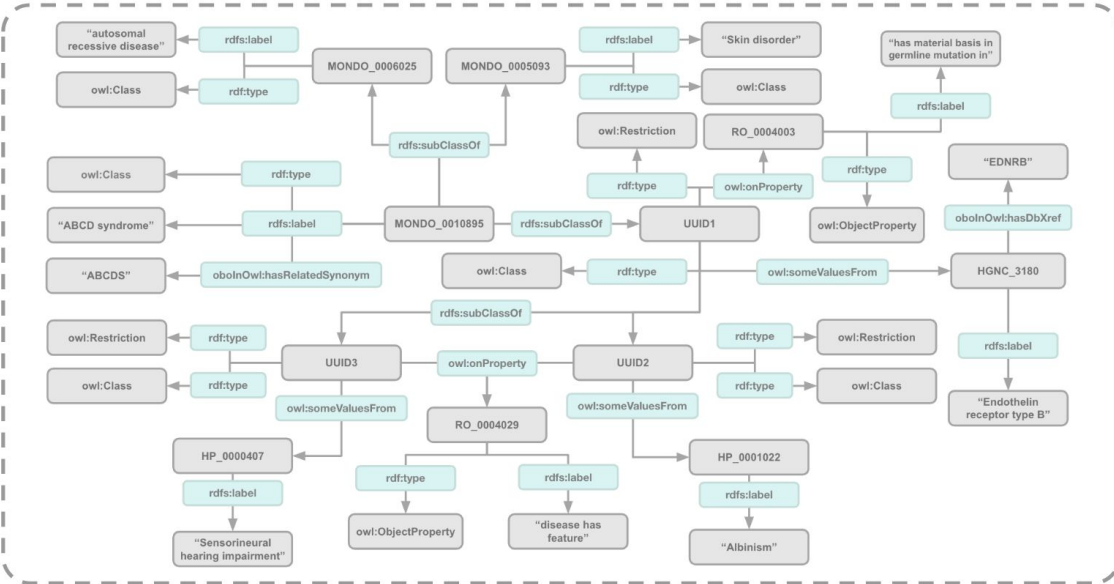
Simple →



← Hybrid



Complex →



## Biological Knowledge Graphs



# Motivation

---

## Design Challenges

- Multiple approaches to modeling biomedical knowledge<sup>1-3</sup>
- No existing benchmarks for knowledges graphs built using different knowledge models

## Implementation Challenges

- Methods may have varying functional, logical, and semantic consequences<sup>1</sup>
- Existing implementations are siloed and complicated
- Semantic Web standard is expressive, but unwieldy and of uncertain benefit<sup>2</sup>

**PheKnowLator**: A framework for building large-scale heterogeneous biomedical knowledge graphs under alternate knowledge models and hub for knowledge graph benchmarks

<https://github.com/callahantiff/PheKnowLator>



## Process Data

### Download

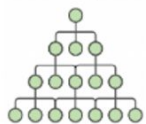
Clinical Records



Experimental Results



Ontologies



Linked Open Data



Knowledge Graphs

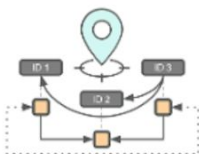


### Preparation

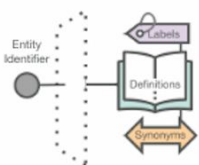
Data Cleaning



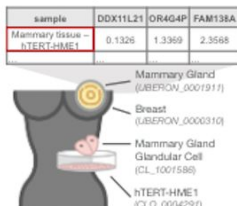
Identifier Mapping



Metadata Generation



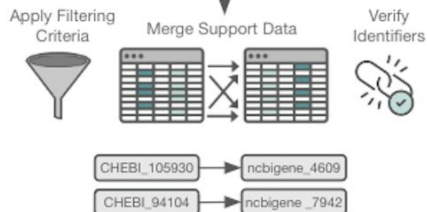
Concept Annotation



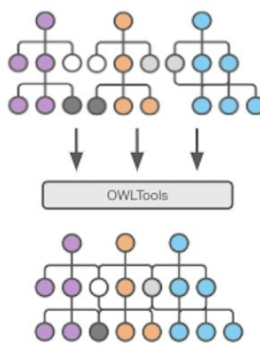
## Construct Knowledge Graphs

### Edge List Construction

chemicalID	Symbol	GeneForms	Organism	PubMed
C534883	MYC	protein	mus musculus	21344345
C534883	MYC	protein	homo sapiens	32184358
C553100	TFEB	mRNA	homo sapiens	25716159
C553100	TFEB	protein	homo sapiens	26474267



### Ontology Alignment

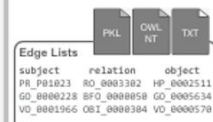


### Output Generation

Quality/Provenance Reports, Logs

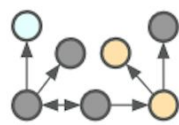


Knowledge Graphs



## Customize Knowledge Representation

Knowledge Model

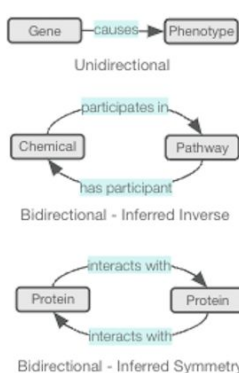


Instance-based

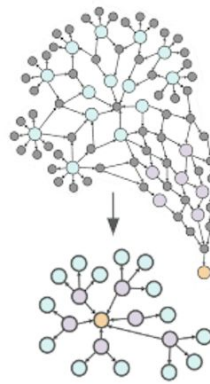


Class-based

Relation Strategy



Semantic Abstraction

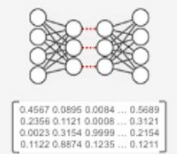


## Benchmarks

Knowledge Graphs



Embeddings



## Tools

Jupyter Notebooks



Cloud File Storage



Triplestore



# PheKnowLator Build Types

---

PheKnowLator is capable of building 12 different versions of the same knowledge graph by altering the following:

**Knowledge Model:** the approach used to convert a simple edge list into a rich OWL-based semantic representation.

**Relation Strategy:** the approach used when adding edges to the core set of ontologies.

**Semantic Abstraction:** a lossless compression algorithm that converts a rich OWL-based representation into a simple RDF property graph.

- With or without harmonizing the abstracted knowledge graph to be consistent with a class- or instance-based knowledge model

<https://github.com/callahantiff/PheKnowLator>

# Knowledge Modeling Approaches

**Example:** Add <<EDNRB, Causes, ABCD syndrome>> to an ontologically-grounded knowledge graph.

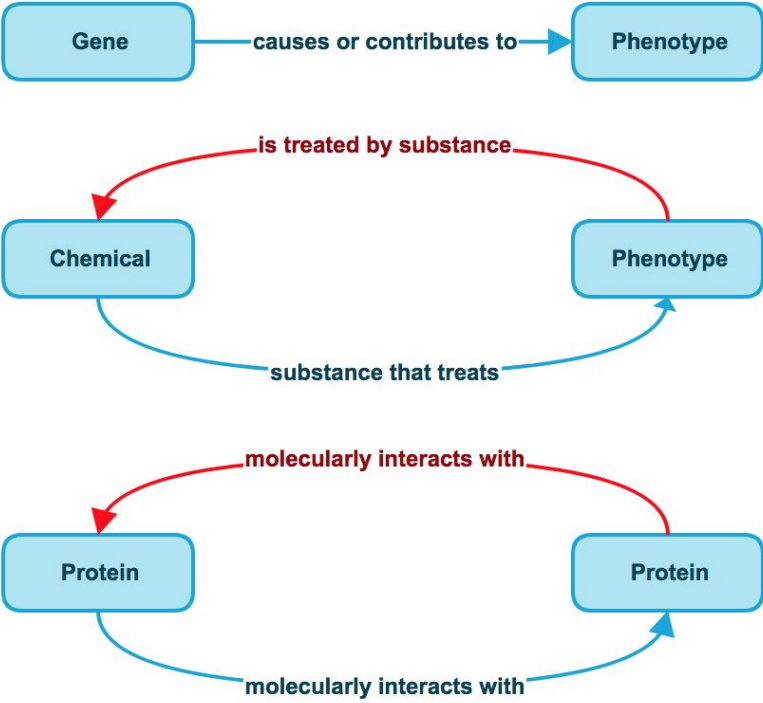
**Challenge:** EDNRB is not currently represented in an ontology. ABCD syndrome is a class in the Human Phenotype Ontology, and is included in the knowledge graph.

**Solution:** Gene is a class in the Sequence Ontology and can be used to add EDNRB to the knowledge graph using two different strategies.

Instance-based Knowledge Model (ABox)	Class-based Knowledge Model (TBox)
<p>EDNRB, rdfs:subClassOf, Gene                      EDNRB, rdf:type, owl:Class</p> <p>UUID1, rdf:type, EDNRB                      UUID1, rdf:type, owl:NamedIndividual</p> <p>UUID2, rdf:type, ABCD syndrome                      UUID2, rdf:type, owl:NamedIndividual</p> <p>UUID1, Causes, UUID2</p>	<p>EDNRB, rdfs:subClassOf, Gene                      EDNRB, rdf:type, owl:Class</p> <p>UUID1, rdfs:subClassOf, EDNRB                      UUID1, rdfs:subClassOf, UUID2                      UUID2, rdf:type, owl:Restriction                      UUID2, owl:someValuesFrom, ABCD syndrome                      UUID2, owl:onProperty, Causes</p>

# Knowledge Modeling Approaches

---



# Semantic Abstraction - OWL-NETS

---

**OWL-NETS** (NEtwork Transformation for Statistical learning) is a computational method that reversibly abstracts Web Ontology Language (OWL)-encoded biomedical knowledge into a more biologically meaningful network representation.

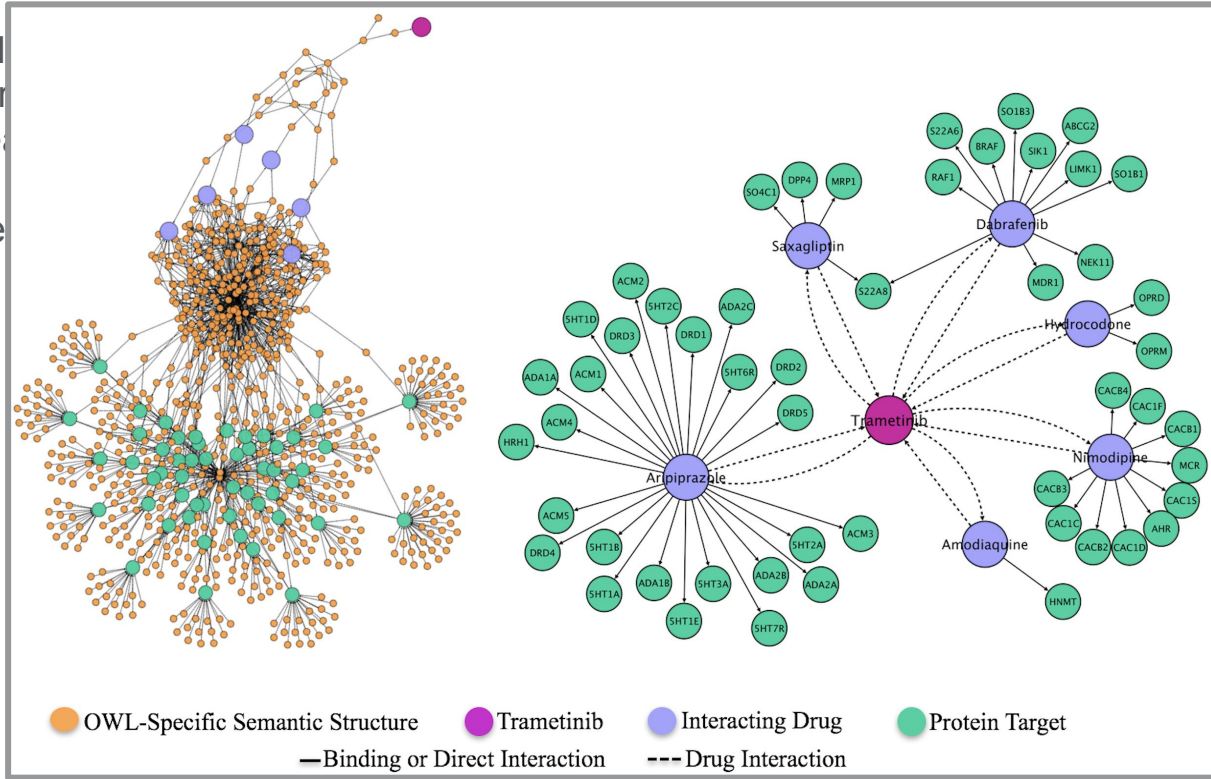
OWL-NETS generates semantically rich knowledge graphs that contain heterogeneous nodes and edges and can be used for tasks that do not require OWL semantics.

<https://github.com/callahantiff/PheKnowLator/wiki/OWL-NETS-2.0>

# OWL-NETS

OWL-NETS (N) reversibly abstract biologically me

OWL-NETS ge edges and can



that  
a more

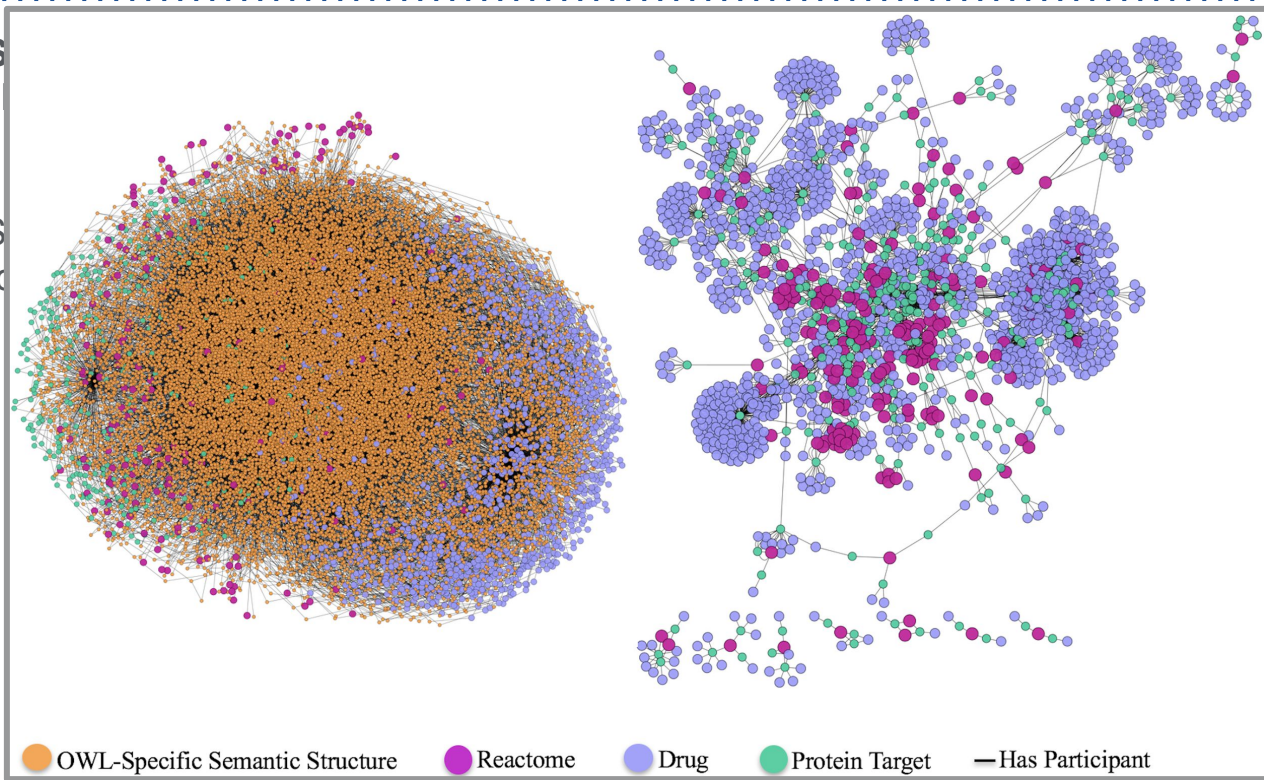
odes and

theKnowLator/wiki/OWL-NETS-2.0

# OWL-NETS

OWL-NETS  
reversibly a  
biologically

OWL-NETS  
edges and c



that  
a more

odes and

[theKnowLator/wiki/OWL-NETS-2.0](https://theKnowLator/wiki/OWL-NETS-2.0)

## Scenario 1: owl unionOf constructor

```
<!-- http://purl.obolibrary.org/obo/CL_0000995 -->
  <owl:Class rdf:about="http://purl.obolibrary.org/obo/CL_0000995">
    <owl:equivalentClass>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://purl.obolibrary.org/obo/CL_0001021"/>
          <rdf:Description rdf:about="http://purl.obolibrary.org/obo/CL_0001026"/>
        </owl:unionOf>
      </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="http://purl.obolibrary.org/obo/CL_0001060"/>
  </owl:Class>
```

[http://purl.obolibrary.org/obo/CL\\_0000995](http://purl.obolibrary.org/obo/CL_0000995)

The OWL class `CL_0000995` (i.e. *CD34-positive, CD38-positive common myeloid progenitor OR CD34-positive, CD38-positive common lymphoid progenitor*) was built by taking the union:

- `CL_0001021` (i.e. *CD34-positive, CD38-positive common lymphoid progenitor*)
- `CL_0001026` (i.e. *CD34-positive, CD38-positive common myeloid progenitor*)

OWL-NETS would decode this class into:

```
CL_0001021, rdfs:subClassOf, CL_0000995
CL_0001026, rdfs:subClassOf, CL_0000995
CL_0000995, rdfs:subClassOf, CL_0001060
```



## The FAIR Guiding Principles

### Findable:

- F1 Data and metadata are assigned a globally unique and persistent identifier
- F2 Data are described with rich metadata (defined by R1 below)
- F3 Metadata clearly and explicitly include the identifier of the data it describes
- F4 Data and metadata are registered or indexed in a searchable resource

### Accessible:

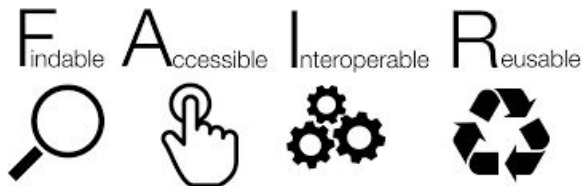
- A1 Data and metadata are retrievable by their identifier using a standardized communications protocol
  - A1.1 The protocol is open, free, and universally implementable
  - A1.2 The protocol allows for an authentication and authorization procedure, where necessary
- A2 Metadata are accessible, even when the data are no longer available

### Interoperable:

- I1 Data and metadata use a formal, accessible, shared, and broadly applicable language for knowledge representation.
- I2 Data and metadata use vocabularies that follow FAIR principles
- I3 Data and metadata include qualified references to other (meta)data

### Reusable:

- R1 Data and metadata are richly described with a plurality of accurate and relevant attributes
  - R1.1 Data and metadata are released with a clear and accessible data usage license
  - R1.2 Data and metadata are associated with detailed provenance
  - R1.3 Data and metadata meet domain-relevant community standards



# Evaluation

---

To evaluate the knowledge graphs that PheKnowLator ecosystem, we conducted the following evaluation tasks:

## Qualitative

- Compare existing open source biomedical knowledge graph construction methods

## Quantitative

- Compare the performance when building the 12 different knowledge graphs designed to represent the molecular mechanisms underlying human disease

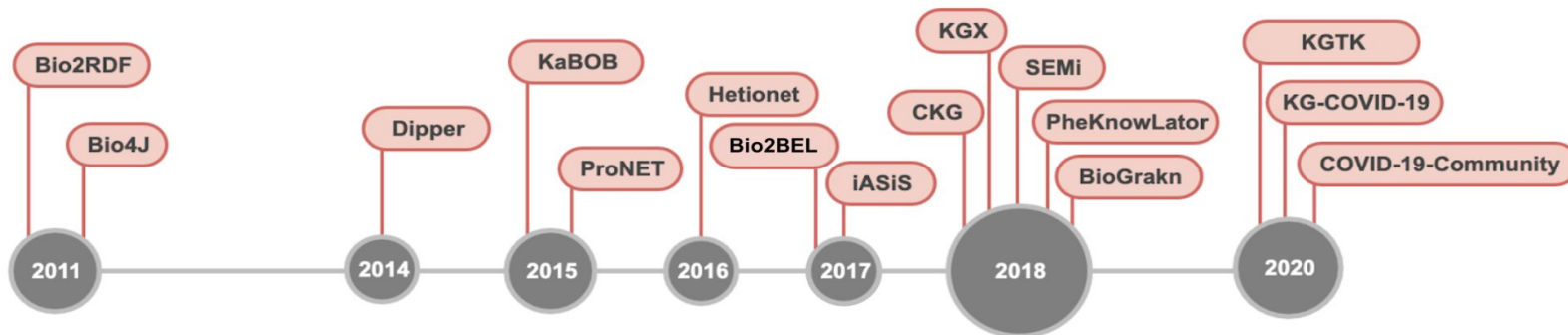
# Evaluation - Comparison to Existing Methods

**Goal:** Compare PheKnowLator to existing open source knowledge graph construction methods

## Identify Similar Methods

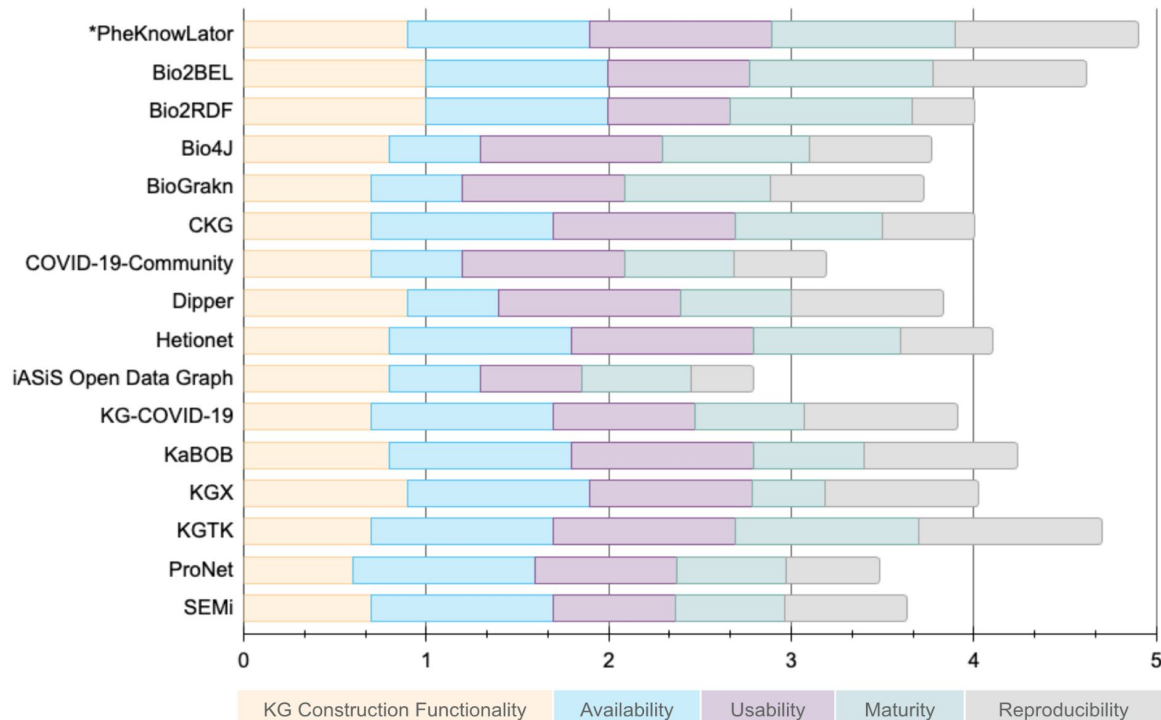
- GitHub API Query
- Recent review articles<sup>1,2</sup>

**Survey:** 5 criteria, 45 questions



# Evaluation - Comparison to Existing Methods

Criteria	Description
Construction Functionality	The steps needed to construct a knowledge graph from downloading and processing data and building edge lists to generating and outputting a KG
Maturity	The level, stage or development phase of a method
Availability	The openness of a method and the ease of obtaining a copy of the method
Usability	Efforts put in place to ensure that a user, with reasonable technical skills, could use the method
Reproducibility	Whether or not the method provides tools or resources to help reproduce the KG construction process and maintain the code base



# Evaluation - Human Disease Mechanism Graph

---

**Goal:** Construct 12 different builds of a knowledge graph designed to represent the molecular mechanisms of human disease and compare them on their build time and memory usage.

## Knowledge Representation

- Designed with PhD molecular biologist
- Validated by wet lab experiments

## Build Data

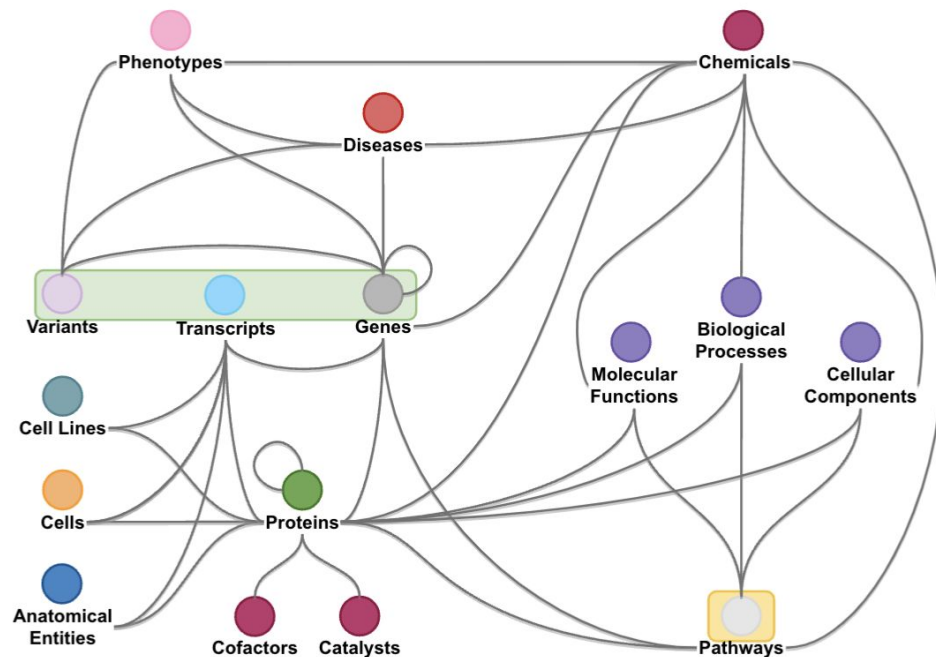
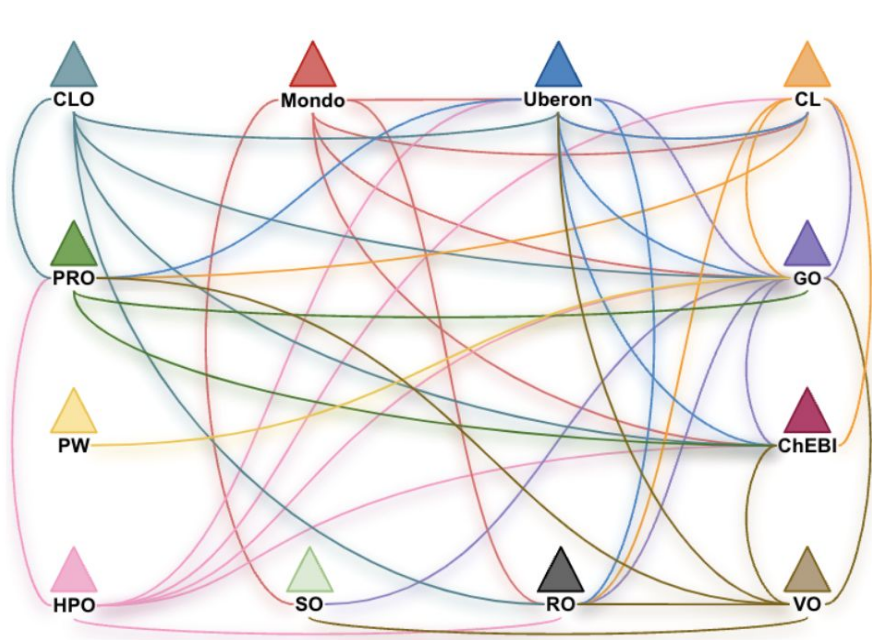
- 12 Open Biological and Biomedical Ontologies Foundry ontologies
- 31 Linked Open Data
- 18 primary node types\*
- 35 primary edge types\*

\*Added to existing node and edge types present in the core set of ontologies

<https://github.com/callahantiff/PheKnowLator>



# Evaluation - Human Disease Mechanism Graph



<https://github.com/callahantiff/PheKnowLator>

# Evaluation - Human Disease Mechanism Graph

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

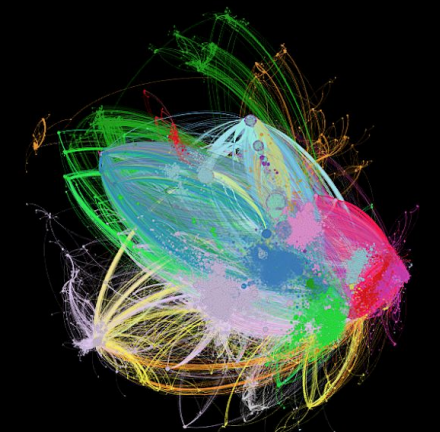
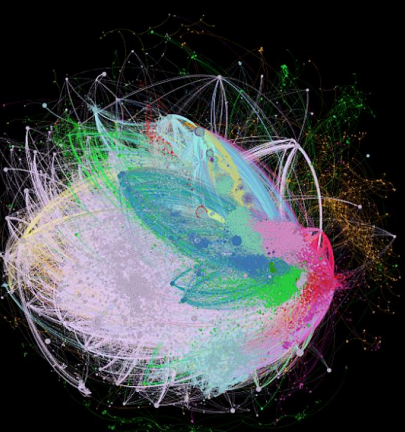
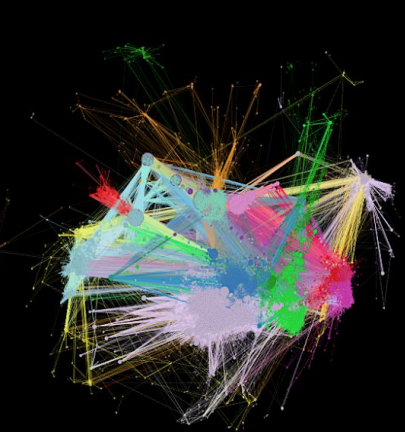
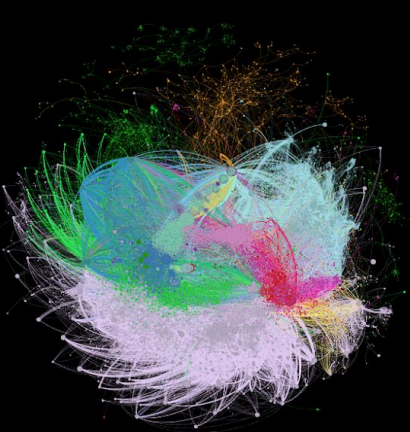
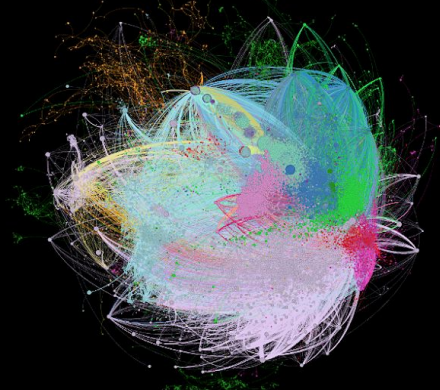
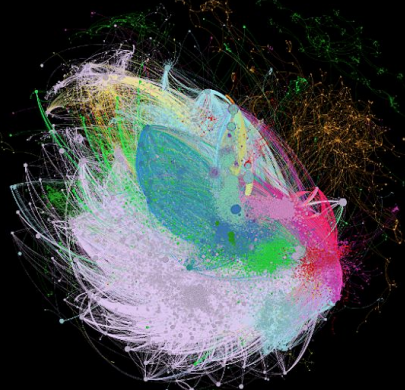
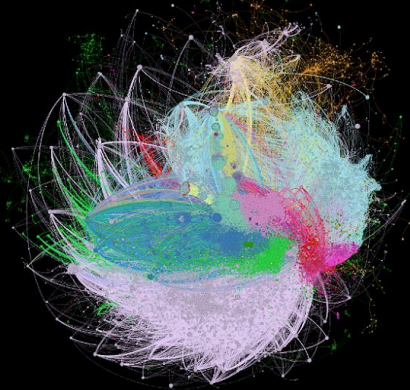


Subclass – Relations Only

Subclass – Relations Only (Harmonized)

Subclass – Inverse Relations

Subclass – Inverse Relations (Harmonized)



Instance – Relations Only

Instance – Relations Only (Harmonized)

Instance – Inverse Relations

Instance – Inverse Relations (Harmonized)

# PheKnowLator Application

---

## Clinical Data: Children's Hospital of Colorado

- 10,000 random medically complex patients ( $\geq 10$  visits)
- 2,646 rare disease patients ( $\geq 10$  visits)

## Embedding Models

- KG Mechanisms
  - PheKnowLator node embeddings<sup>1</sup>
  - Sum and mean embedding aggregation
- Clinical Data
  - Baseline  $\rightarrow$  Bag-of-Words with TF-IDF<sup>3</sup> (**BOW+TF-IDF**)
  - Doc2Vec<sup>2</sup>  $\rightarrow$  Paragraph vector distributed memory (**PV-DM**); Distributed bag of words (**PV-DBOW**)

## Rare Disease Patients

**PKU:** Phenylketonuria (n=235)

**CH:** Congenital Hypothyroidism (n=760)

**SCD:** Sickle Cell Disease (n=816)

**CF:** Cystic Fibrosis (n=835)

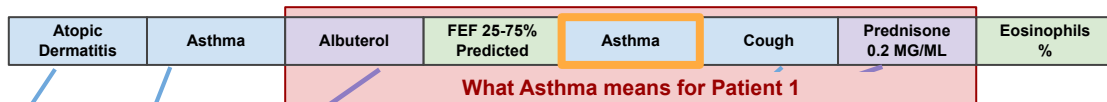
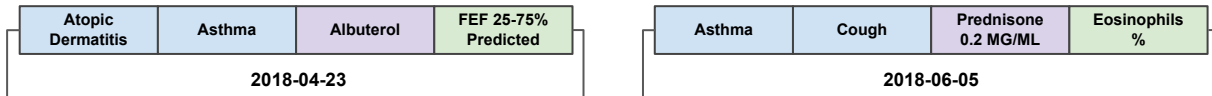
## Evaluation

- One-vs-the-rest multiclass classification
- Random forest, SVM, Naïve Bayes

## EHR Database

Date	Event	Event Label
2018-04-23	ICD10CM: L20.9	Atopic Dermatitis
2018-04-23	SNOMED-CT: 195967001	Asthma
2018-04-23	RXNORM: 435	Albuterol
2018-04-23	LOINC: 69971-0	FEF 25-75% Predicted
2018-06-05	SNOMED-CT: 195967001	Asthma
2018-06-05	SNOMED-CT: 263731006	Cough
2018-06-05	RXNORM: 335528	Prednisone 0.2 MG/ML
2018-06-05	LOINC: 26450-7	Eosinophils %
...	Data for Patient 1	...

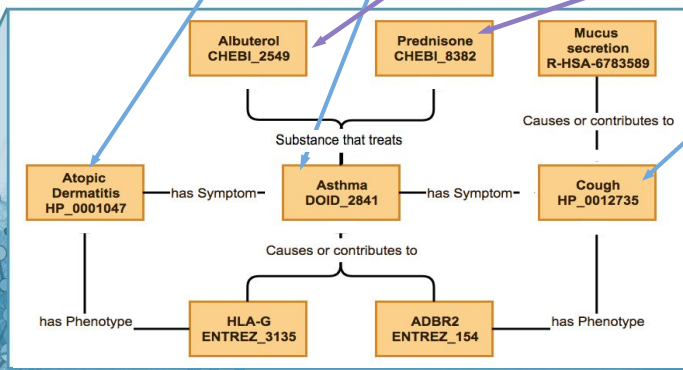
## Creating Representations Using Clinical Data



Patient 1's Clinical Embedding = [0.047389 -0.037575 -0.033601 0.04825922... -0.05467]

Inside EHR

Outside EHR



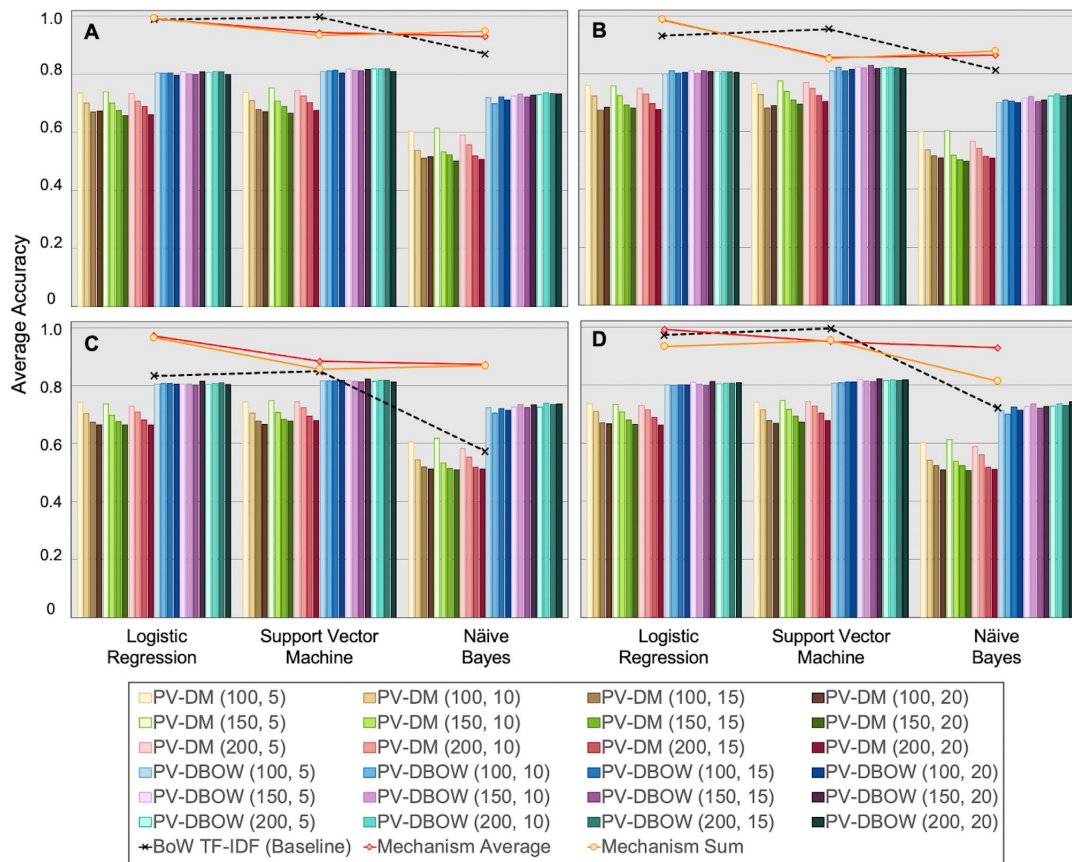
**DOID\_2841** = [0.025812 -0.047533 -0.000807 -0.026119 ... 0.03369]  
**HP\_0012735** = [0.002039 0.007075 -0.144156 0.100861 ... -0.526781]  
**HP\_0001047** = [0.078954 0.0041178 0.112156 -0.111861 ... 0.874445]  
**CHEBI\_8382** = [-0.079752 0.056994 0.023846 -0.006980 ... 0.578942]  
**CHEBI\_2549** = [0.074613 -0.003261 0.054199 -0.005925 ... 0.002140]  
**ENTREZ\_154** = [-0.046503 0.045141 -0.008021 0.002155 ... 0.433895]  
**ENTREZ\_3135** = [0.031590 -0.005769 -0.026917 -0.002815 ... 0.018874]  
**R-HSA-6783589** = [-0.011119 -0.111420 0.013410 -0.041020 ... -0.047533]

Patient 1's Transformed Clinical Codes = [HP\_0012735 HP\_0031352 CHEBI\_8382 CHEBI\_2549 DOID\_284 ... HP\_0012735]

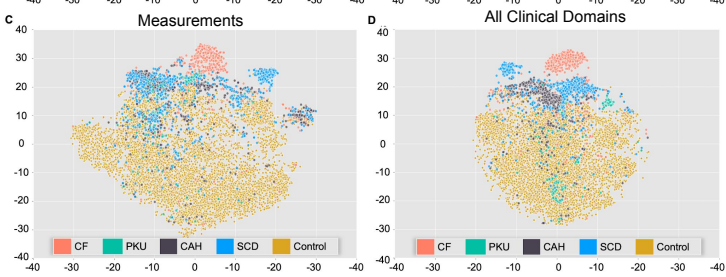
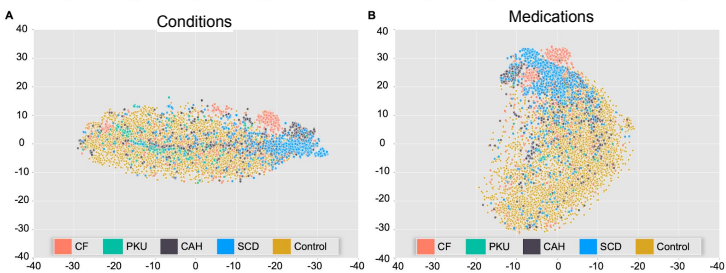
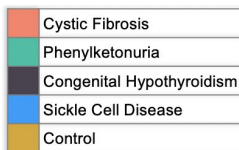
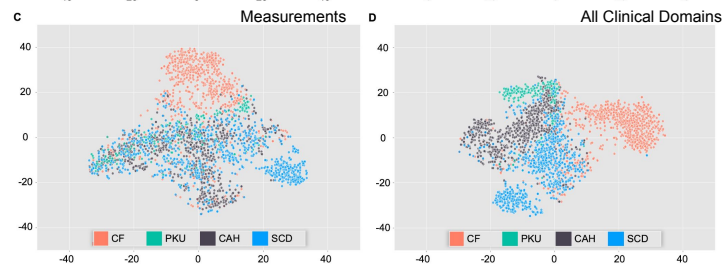
Patient 1's Mechanism Embeddings = [[0.002039...], [0.078954...], [-0.079752...], [0.074613...], [0.025812...], ... [0.002039...]]

## Creating Representations Using Estimates of Molecular Mechanisms

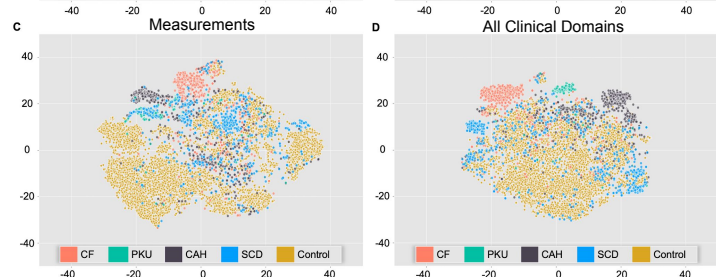
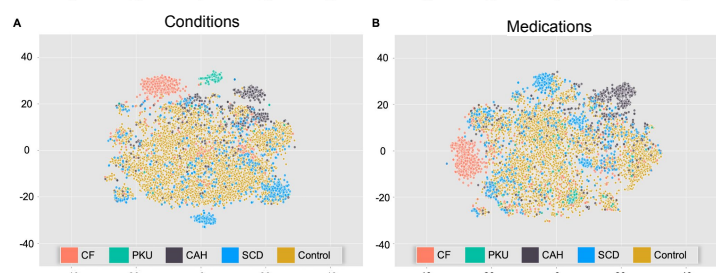
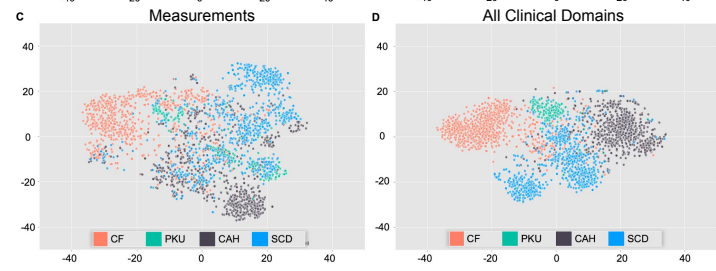
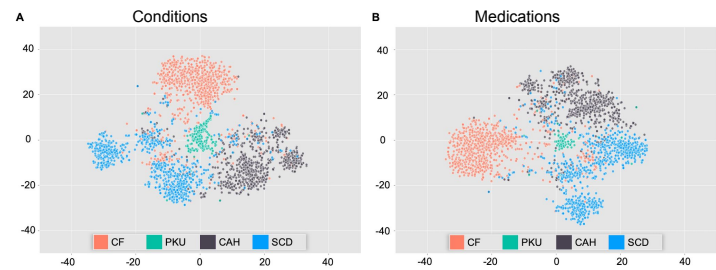
# PheKnowLator Application



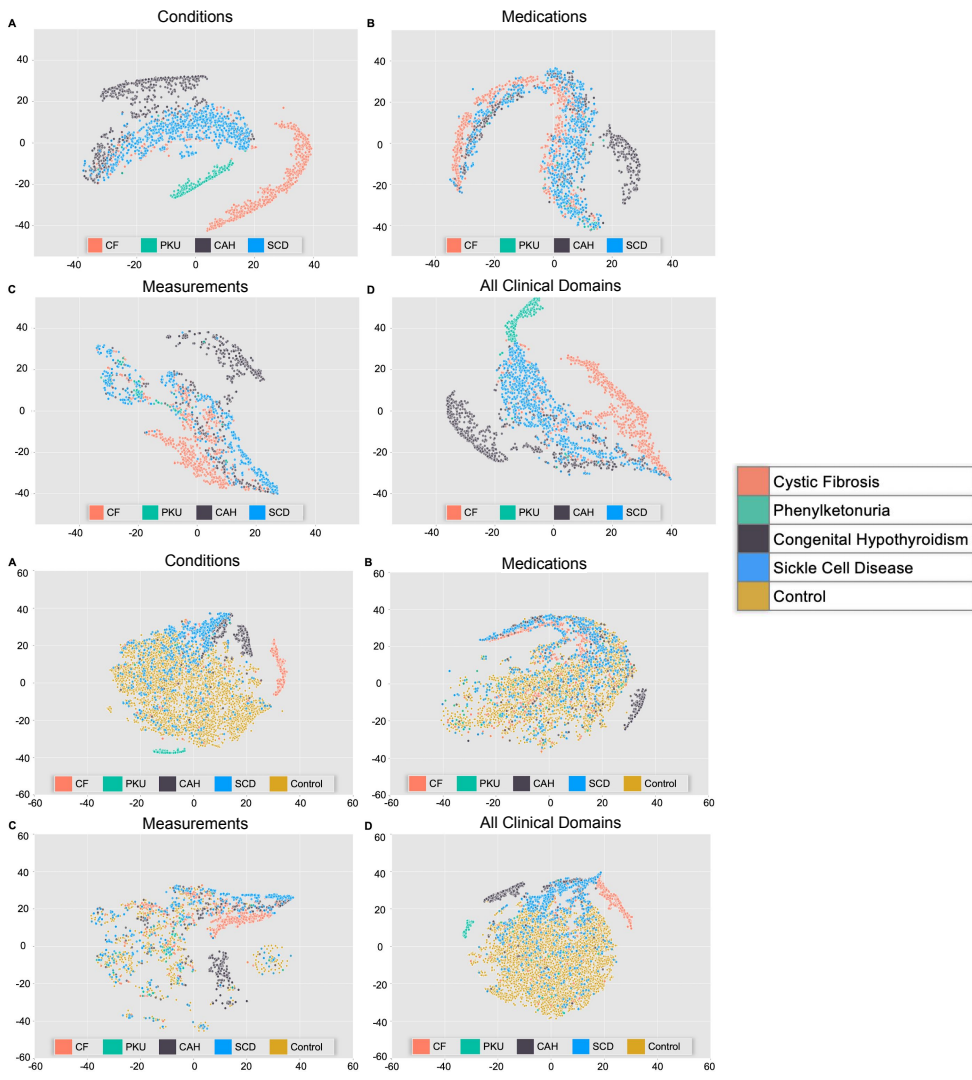
Best Clinical Model: PV-CBOW



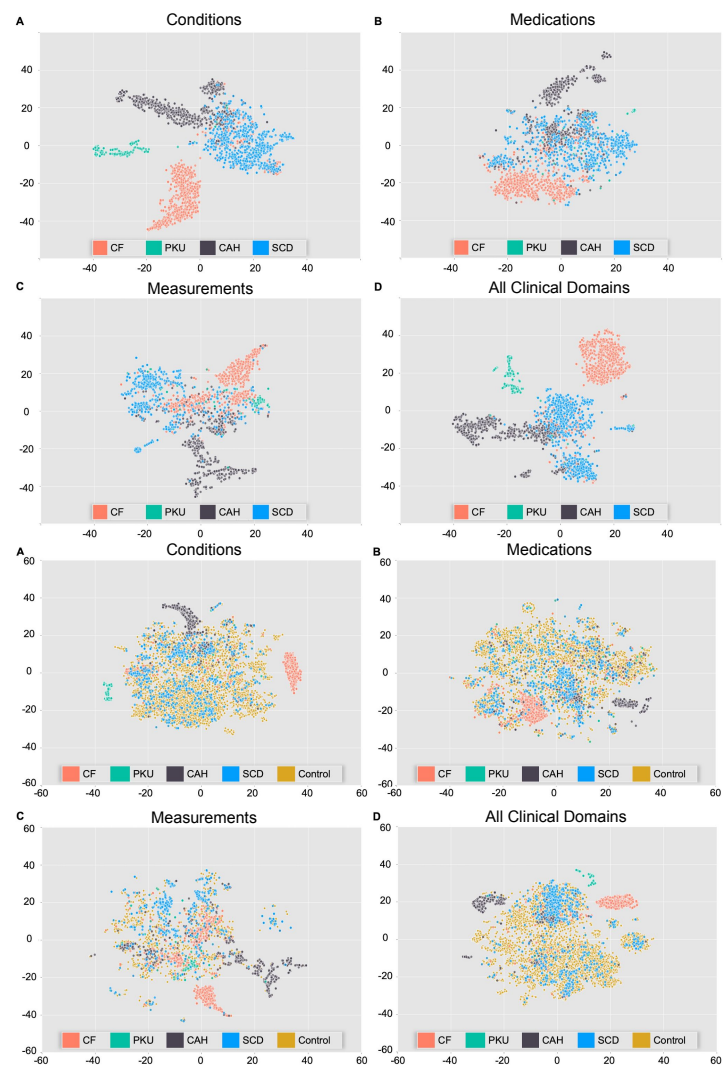
Best Clinical Model: PV-DBOW



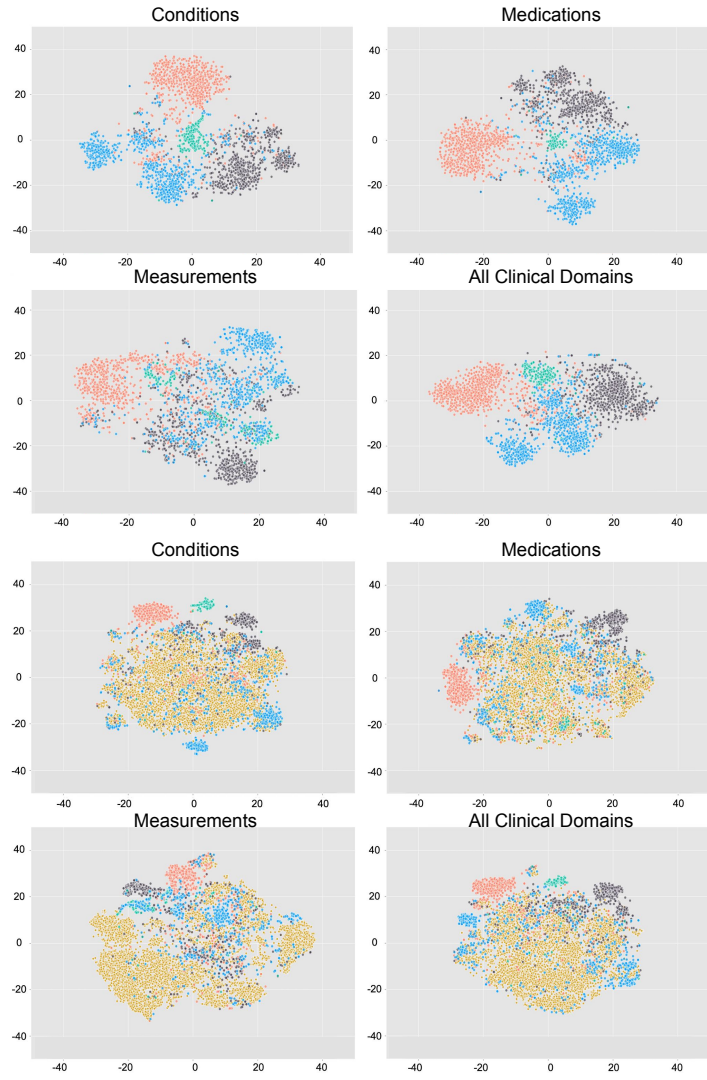
Best Knowledge Graph Model: Summed Node Embeddings



Best Knowledge Graph Model: Averaged Node Embeddings



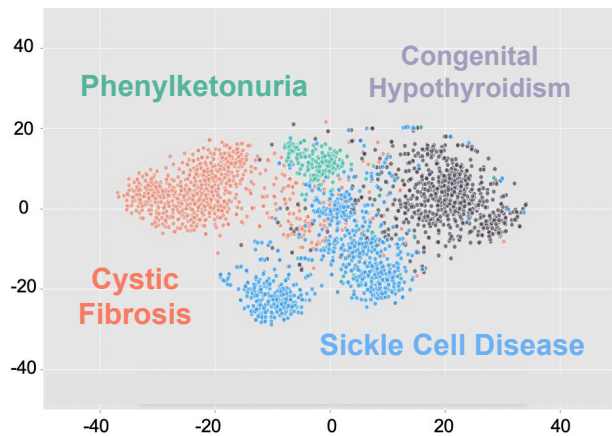
Best Clinical Data Model (PV-CBOW)



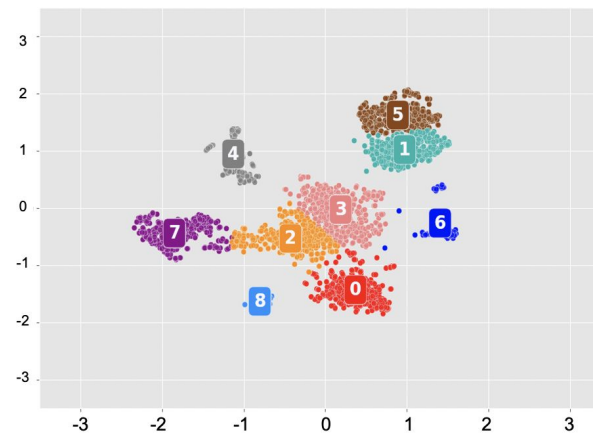
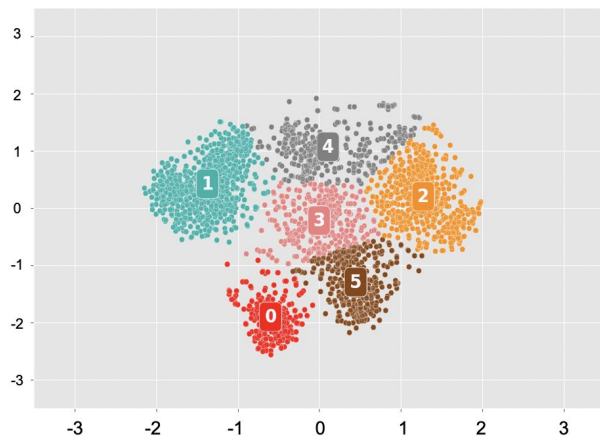
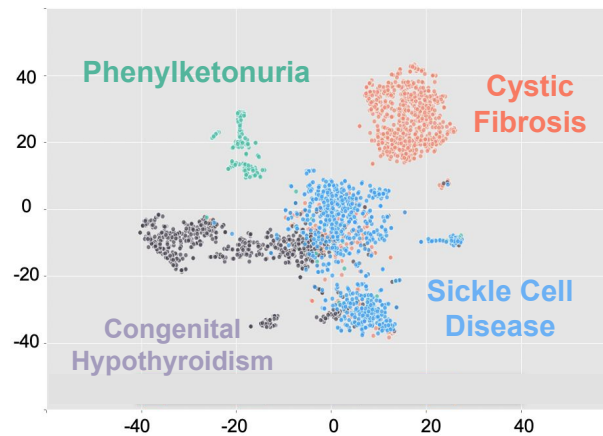
Best Knowledge Graph Model (averaged node embedding)



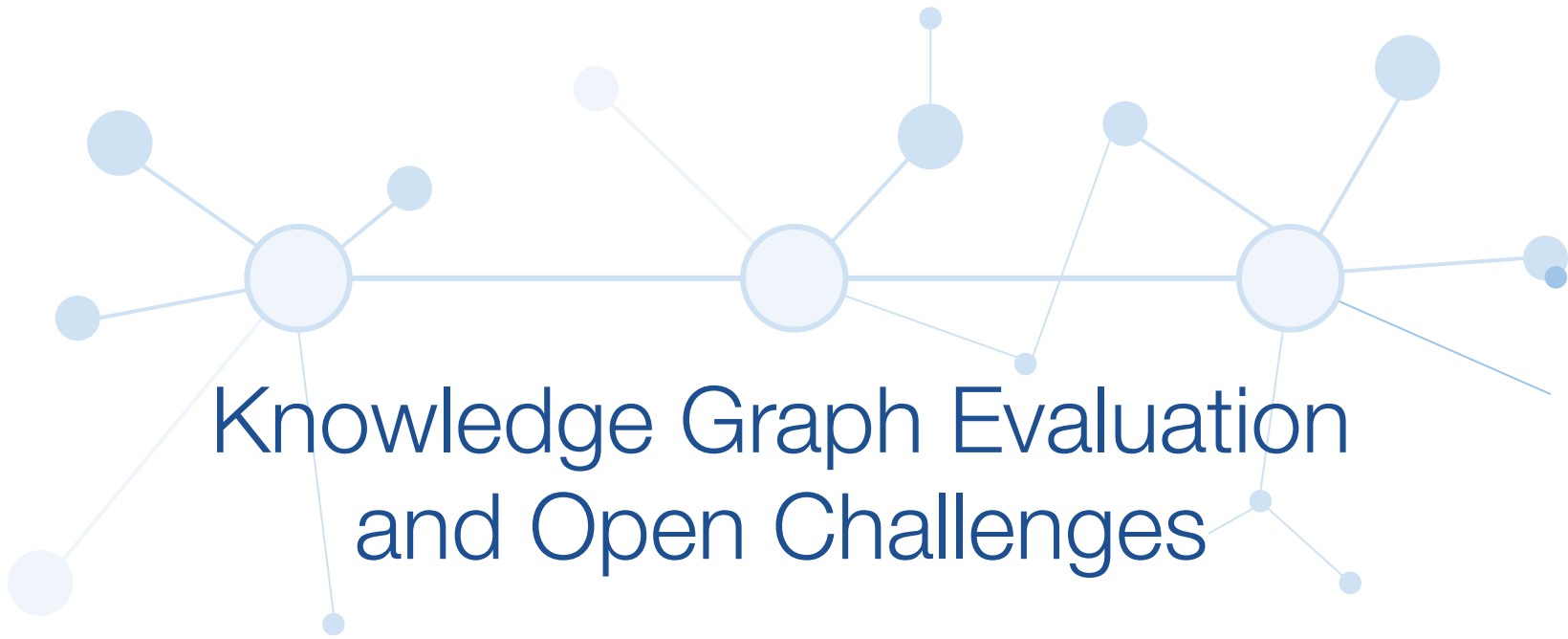
Best Clinical Data Model



Best Knowledge Graph Model







## Knowledge Graph Evaluation and Open Challenges

# Evaluating Biomedical Knowledge Graphs

---

Knowledge graph evaluation aims to prove that a resulting graph is "good". When using formal semantics, this can be done (mostly) with a description logics reasoner.

# Evaluating Biomedical Knowledge Graphs

---

Knowledge graph evaluation aims to demonstrate that a resulting graph is accurate. When using formal semantics, this can be done (mostly) with a description logics reasoner.

Evaluating a knowledge graph also includes verifying its data quality. This can be very challenging given the scope and heterogeneity of knowledge graphs. Although research is limited, existing work has used the following metrics:

- Percentage of triples in the knowledge graph that are correct<sup>1</sup>
  - Given a triple, the corresponding relationship is consistent with a real-life fact as judged by a human (randomly sampled)
- Accuracy, timeliness, trustworthiness, consistency, completeness, and availability<sup>2</sup>
- Precision/recall when using the knowledge graph to predict specific edge types that are known or domain expert validation of unknown or edge types or predictions

# Open Challenges - Part 1

---

**Computational performance:** biomedical knowledge is very extensive, which can result in biomedical knowledge graphs that contain billions of assertions

# Open Challenges - Part 1

---

**Computational performance:** biomedical knowledge is very extensive, which can result in biomedical knowledge graphs that contain billions of assertions

**Knowledge graph construction:** a few knowledge graphs are constructed through intense expert curation. Automated approaches to knowledge graph construction fall into two broad classes:

- **Data-driven** knowledge graph construction can involve the integration of previously disparate resources or the direct analysis of large-scale datasets
- **NLP** methods are imperfect and often focused on assessing the reliability of extracted information or on techniques to manage missing or erroneous assertions and other noise

# Open Challenges - Part 1

---

**Computational performance:** biomedical knowledge is very extensive, which can result in biomedical knowledge graphs that contain billions of assertions

**Knowledge graph construction:** a few knowledge graphs are constructed through intense expert curation. Automated approaches to knowledge graph construction fall into two broad classes:

- **Data-driven** knowledge graph construction can involve the integration of previously disparate resources or the direct analysis of large-scale datasets
- **NLP** methods are imperfect and often focused on assessing the reliability of extracted information or on techniques to manage missing or erroneous assertions and other noise

**Data integration:** There are thousands of public, biomedically important databases; hence, integration approaches that support semantic compatibility are important but rarely out-of-the-box able to be integrated

# Open Challenges - Part 2

---

**The knowledge acquisition bottleneck:** human curation is difficult to scale. Alternatives to manual curation, including applications of text mining and machine learning, have shown promise, but are still far short of human-like performance

# Open Challenges - Part 2

---

**The knowledge acquisition bottleneck:** human curation is difficult to scale. Alternatives to manual curation, including applications of text mining and machine learning, have shown promise, but are still far short of human-like performance

**How to represent what is not known:** any scientist can describe gaps, ambiguities and uncertainties in existing knowledge, yet there are few computational methods capable of representing, let alone reasoning about, ignorance



# Open Challenges - Part 2

---

**The knowledge acquisition bottleneck:** human curation is difficult to scale. Alternatives to manual curation, including applications of text mining and machine learning, have shown promise, but are still far short of human-like performance

**How to represent what is not known:** any scientist can describe gaps, ambiguities and uncertainties in existing knowledge, yet there are few computational methods capable of representing, let alone reasoning about, ignorance

**Inference methods are lacking:** existing inference methods are far short of the range and creativity of human experts in developing potential explanations, generating significant hypotheses, and generally interpreting results in light of previous knowledge. Promising inference methods scale poorly and are constrained in their ability to harness large knowledge-bases by the extremely large computational loads involved

# Part 2 - Activity



Overview

# Exercise Instructions

---

## Set-up Environment

Clone the PheKnowLator GitHub: `git clone https://github.com/callahantiff/PheKnowLator.git`

Use the URL below to download the zipped directory, which is called `kg_course_block3_exercise`, to the cloned PheKnowLator directory and unzip so its location is:

`PheKnowLator/kg_course_block3_exercise` → <https://tinyurl.com/kgblock3>

## Exercise Overview

The goal of this exercise is to provide exposure and experience with building and exploring different types of knowledge graphs. The notebook will give you experience with the following tasks:

1. Implementing different strategies for constructing knowledge graph edge lists
2. Performing semantic abstraction on an ontology
3. Constructing a knowledge graph using functionality from the PheKnowLator Ecosystem