

Ubergraph: integrating OBO ontologies into a unified semantic graph

Jim Balhoff (RENCI, UNC Chapel Hill)

Ugur Bayindir, Anita Caron, David Osumi-Sutherland (EBI)

Nico Matentzoglou (Semanticly)

Chris Mungall (LBNL)

ICBO 2022—Ann Arbor, Michigan

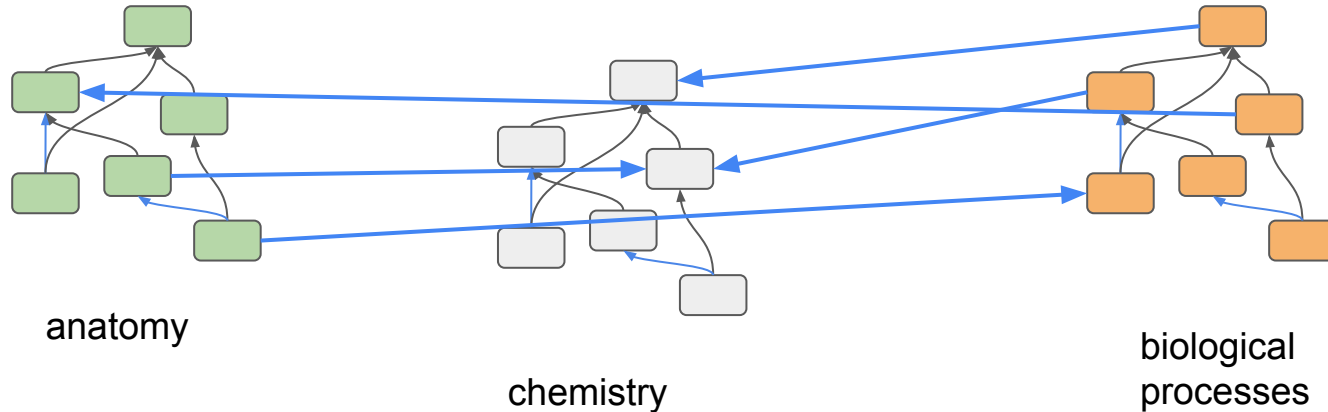
What is Ubergraph?

- **A particular online resource:**
 - RDF knowledge graph of OBO ontologies (39 so far)
 - Public SPARQL query endpoint
- **An approach to rendering OWL ontologies as knowledge graphs:**
 - Ergonomic traversal and query
 - Access to class and relation semantics

OBO Knowledge Graph

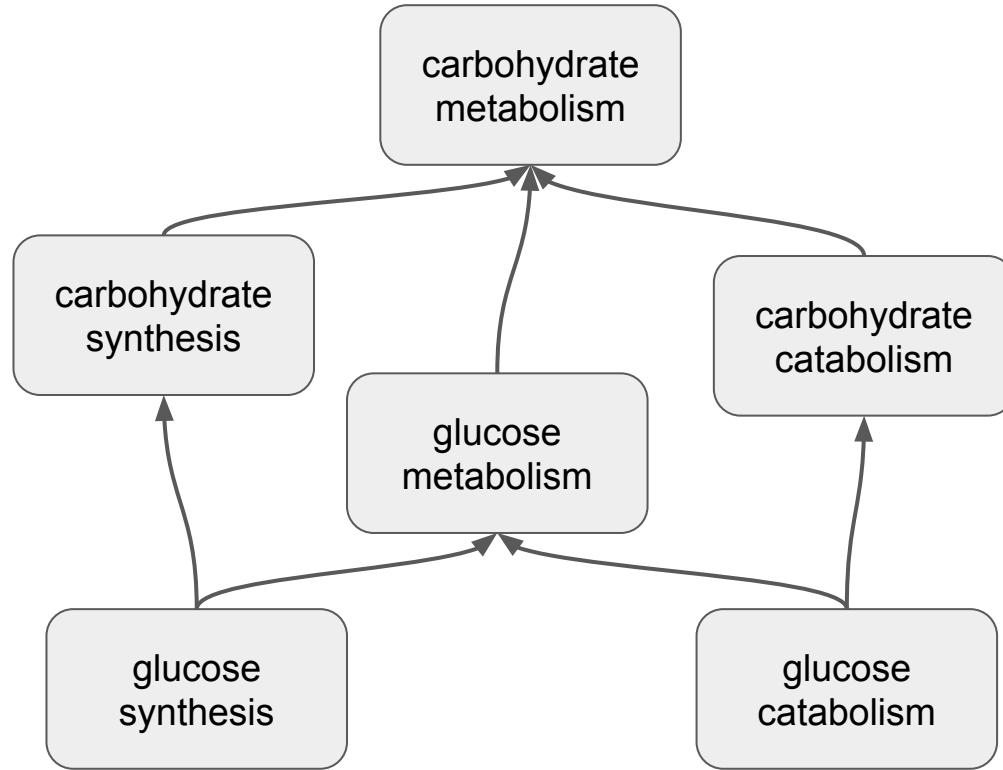
Why “knowledge graph” and not “ontology repository”?

OBO mission: *“develop a family of interoperable ontologies that are both logically well-formed and scientifically accurate”*



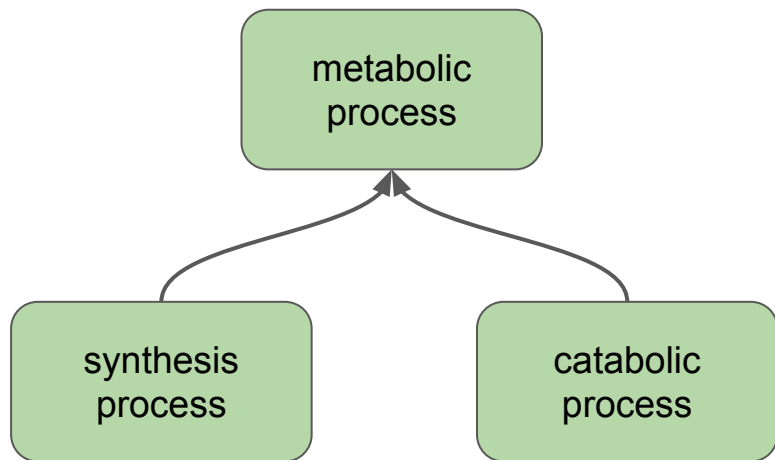
OBO ontologies are mutually referential

Concept reuse...

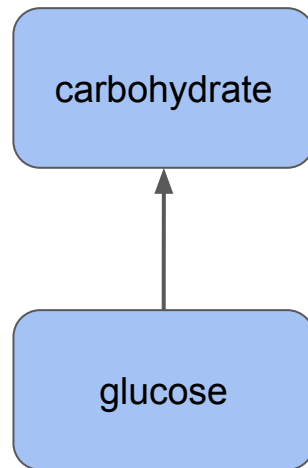


Gene Ontology

Concept reuse



Gene Ontology



CHEBI

Core hierarchies

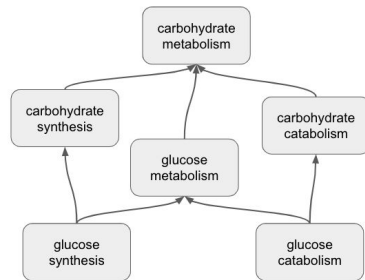
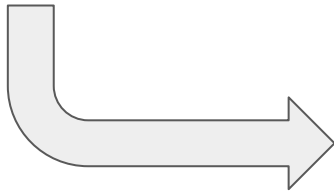
Concept reuse

Standing on the shoulders of giants:

- 'carbohydrate metabolism' == 'metabolic process' and 'has input or output' some 'carbohydrate'
- 'carbohydrate synthesis' == 'synthesis process' and 'has output' some 'carbohydrate'
- 'carbohydrate catabolism' == 'catabolic process' and 'has input' some 'carbohydrate'
- 'glucose metabolism' == 'metabolic process' and 'has input or output' some 'glucose'
- 'glucose synthesis' == 'synthesis process' and 'has output' some 'glucose'
- 'glucose catabolism' == 'catabolic process' and 'has input' some 'glucose'



OWL reasoner

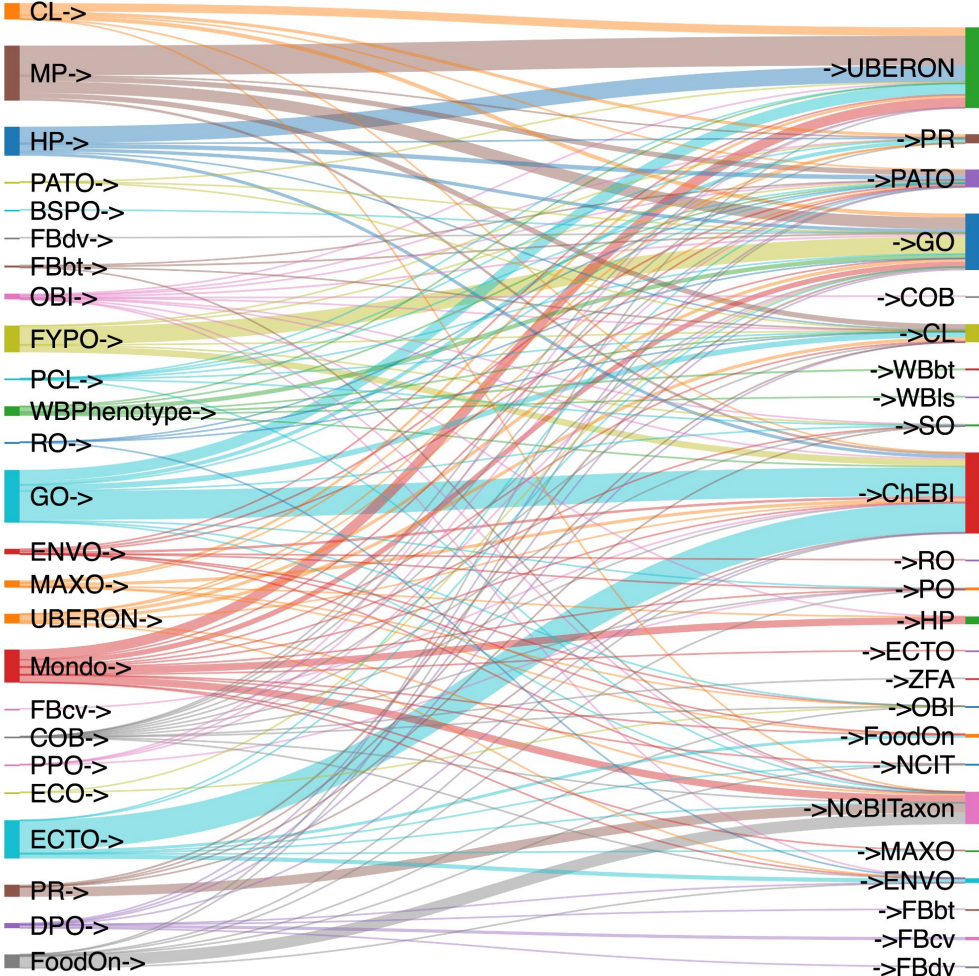


Concept reuse (another example)

HP:Astrocytosis EquivalentTo *RO:has_part* some (**PATO:increased_rate** and (*RO:inheres_in_part_of* some (**GO:cell_growth** and (*RO:occurs_in* some **CL:astrocyte**))) and (*RO:has_modifier* some **PATO:abnormal**)

*The actual logical classification of **Astrocytosis** in HP depends on what is said about all these other terms in the other ontologies... and how other HP terms are defined using related terms.*

Ontologies
with axioms
referencing
terms from
other
ontologies



Ontologies
with terms
referenced in
other
ontologies'
axioms

Merged ontology knowledge graph 👍

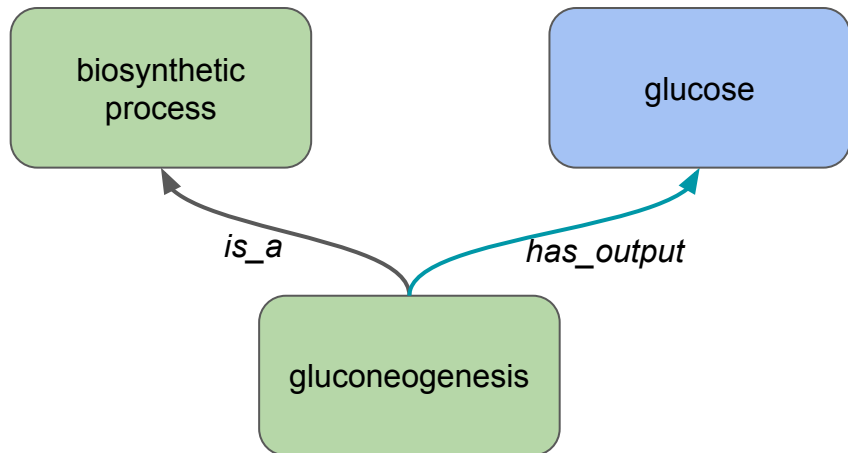
Convenient: OWL ontologies can be natively saved as RDF & stored in a triplestore database

But there are challenges:

- Syntactic/Structural
 - Querying & traversing ontology “source code” is not user friendly
- Semantic
 - Many graph connections are only logically implied, not stated


Problem: querying ontology syntax

```
[Term]  
id: GO:0006094  
name: gluconeogenesis  
def: "The formation of glucose from  
noncarbohydrate precursors, such as pyruvate,  
amino acids and glycerol."  
is_a: GO:0009058 ! biosynthetic process  
relationship: has_output CHEBI:17234 ! glucose
```




SPARQL:

What processes output glucose?



```
SELECT ?process  
WHERE {  
  ?process has_output: CHEBI:17234 .  
}
```



```
SELECT ?process  
WHERE {  
  ?process rdfs:subClassOf ?x .  
  ?x rdf:type owl:Restriction .  
  ?x owl:onProperty has_output: .  
  ?x owl:someValuesFrom CHEBI:17234 .  
}
```



Problem: querying ontology syntax (2)

'gluconeogenesis' EquivalentTo (**'biosynthetic process'** and (*has output* some **'glucose'**)



[Term]
id: GO:0006094
name: gluconeogenesis
def: "The formation of glucose from noncarbohydrate precursors, such as pyruvate, amino acids and glycerol."
intersection_of: GO:0009058 ! biosynthetic process
intersection_of: has_output CHEBI:17234 ! glucose

SPARQL:

```
SELECT ?process
WHERE {
  ?process owl:equivalentClass ?equiv .
  ?equiv rdf:type owl:Class .
  ?equiv owl:intersectionOf ?operands .
  ?operands rdf:rest*/rdf:first ?x .
  ?x owl:onProperty has_output: .
  ?x owl:someValuesFrom CHEBI:17234 .
}
```



Problem: querying ontology semantics

What processes input or output a carbohydrate?

Relation Ontology:

- *'has input' SubPropertyOf 'has participant'*
- *'has output' SubPropertyOf 'has participant'*

```
SELECT ?process
WHERE {
  ?process rdfs:subClassOf ?x .
  ?x rdf:type owl:Restriction .
  ?x owl:onProperty ?property .
  ?property rdfs:subPropertyOf* has_participant: .
  ?x owl:someValuesFrom ?target .
  ?target rdfs:subClassOf* CHEBI:16646 .
}
```

But what about...

Relevant property chains?

- *'has part' o 'has participant' SubPropertyOf 'has participant'*
- *'ends with' o 'has output' SubPropertyOf 'has output'*
- *'starts with' o 'has input' SubPropertyOf 'has input'*

Transitive properties?

We really should be using a reasoner!

DL query:

'has participant' some carbohydrate

Simplified querying: syntax & semantics

Reasoning in Protégé: high memory, slow startup (download ontologies, read, classify...)

Some triplestores incorporate RDFS or OWL reasoners: typically focused on instance classification (not terminology reasoning)

Ubergraph solution: precomputed “**relation graph**”

- Materialize every implied edge between every named class, as simple RDF triples
 - $A \text{ SubClassOf } B \Rightarrow A \text{ rdfs:subClassOf } B$
 - $A \text{ SubClassOf } (r \text{ some } B) \Rightarrow A \text{ r } B$

gluconeogenesis relation graph

- `'gluconeogenesis' rdfs:subClassOf 'biosynthetic process'`
- `'gluconeogenesis' rdfs:subClassOf 'metabolic process'`
- `'gluconeogenesis' rdfs:subClassOf ... 'biological process'`
- `'gluconeogenesis' 'has output' 'glucose'`
- `'gluconeogenesis' 'has output' 'carbohydrate'`
- `'gluconeogenesis' 'has output' ... 'chemical entity'`
- `'gluconeogenesis' 'has participant' 'glucose'`
- `'gluconeogenesis' 'has participant' ... chemical entity'`
- *and many more*

relation-graph

- Open source tool for computing ontology closures
- Based on Whelk reasoner
 - OWL EL reasoner providing fast, parallel answering of DL queries
- Computes every subclass of $r \text{ some } C$ for all combinations of properties (r) and classes (C) in the ontology
 - Outputs triple $X \text{ } r \text{ } C$ for every query result X
- Parallelized algorithm; avoids unnecessary queries
- Streaming output to RDF N-triples
- <https://github.com/balhoff/relation-graph>
- Being integrated into a ROBOT command

Querying the relation graph

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX CHEBI: <http://purl.obolibrary.org/obo/CHEBI_>
PREFIX has_output: <http://purl.obolibrary.org/obo/R0_0002234>
SELECT ?process ?label
WHERE {
  ?process has_output: CHEBI:17234 .
  ?process rdfs:label ?label .
}
```

Which processes
output glucose?

https://api.triplydb.com/s/ltC_U-kC7

?process	?label
GO:0006094	gluconeogenesis
GO:0019574	sucrose catabolic process via 3'-ketosucrose

Querying the relation graph

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX adrenal_gland: <http://purl.obolibrary.org/obo/UBERON_0002369>
PREFIX part_of: <http://purl.obolibrary.org/obo/BFO_0000050>
SELECT DISTINCT ?x ?x_label
WHERE {
  adrenal_gland: part_of: ?x .
  ?x rdfs:label ?x_label .
}
```

What is the adrenal gland part of?

<https://api.triplydb.com/s/joa25tMxs>

?x	?x_label
UBERON:0000916	abdomen
UBERON:0010074	chromaffin system
UBERON:0001062	anatomical entity
BFO:0000004	independent continuant

+34

Querying the relation graph

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX cell: <http://purl.obolibrary.org/obo/CL_0000000>
PREFIX organ: <http://purl.obolibrary.org/obo/UBERON_0000062>
PREFIX abdomen: <http://purl.obolibrary.org/obo/UBERON_0000916>
PREFIX part_of: <http://purl.obolibrary.org/obo/BFO_0000050>
SELECT DISTINCT ?cell ?cell_label ?organ ?organ_label
WHERE {
  ?cell rdfs:subClassOf cell: .
  ?cell part_of: ?organ .
  ?organ rdfs:subClassOf organ: .
  ?organ part_of: abdomen: .
  ?cell rdfs:label ?cell_label .
  ?organ rdfs:label ?organ_label .
}
```

Which cell types are
parts of which organs
in the abdomen?

<https://api.triplydb.com/s/bJQu3rRtM>

?cell	?cell_label	?organ	?organ_label
CL:0002523	mesonephric podocyte	UBERON:0000080	mesonephros
CL:2000051	splenic fibroblast	UBERON:0002106	spleen

+387

Querying the relation graph

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX bone_element: <http://purl.obolibrary.org/obo/UBERON_0001474>
PREFIX skeletal_system: <http://purl.obolibrary.org/obo/UBERON_0001434>
PREFIX part_of: <http://purl.obolibrary.org/obo/BFO_0000050>
ASK WHERE {
  ?bone rdfs:subClassOf bone_element:
  FILTER NOT EXISTS { ?bone part_of: skeletal_system: . }
}
```

Check entailments:
are there any bone
elements not inferred
to be part of the
skeletal system?

<https://api.triplydb.com/s/iPgdyqWiT>

Nope!



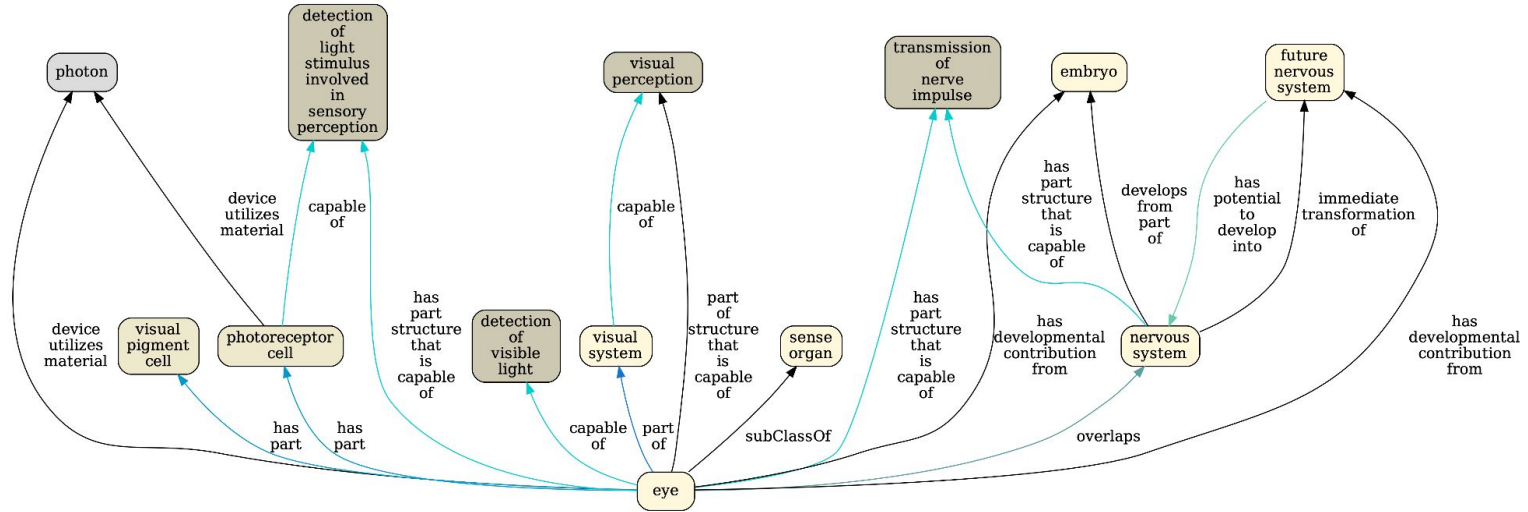
Traversing the knowledge graph incrementally

“Nonredundant” graph

- Separate named graph containing only “direct” edges
- Pruning rules remove redundant edges:
 - Any edges that can be recapitulated via transitive SPARQL property path: `rdfs:subClassOf`, any object properties of type `owl:TransitiveProperty`
 - Any edges with same subject and predicate as one with a more specific object
 - Any edges with same predicate and object as one with a more general subject
 - Any edges with same subject and object as one with a more specific predicate
- Convenient for ontology browsing UIs, term info, etc.
- Pruning is accomplished using a small Soufflé Datalog script on the full relation-graph output

Terms one hop from 'eye' (outgoing)

Edges in the nonredundant graph



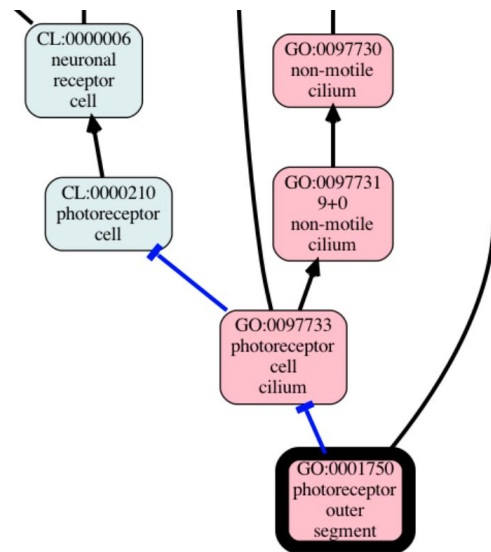
Ubergraph specifics

- **39** OBO ontologies (currently, growing outward from Uberon)
 - Including ChEBI, NCBITaxon, PRO
- **530,834,705** triples
 - complete relation graph: **318,231,131** triples
 - ~4 hours to compute
 - nonredundant relation graph: **4,532,758** triples
 - ~1 hour to compute
- Automated Makefile build (and docs) at <https://github.com/INCATools/ubergraph>
- Deployed using the Blazegraph triplestore in an on-premises Kubernetes cluster at RENCI

Accessing Ubergraph

- **Public SPARQL endpoint** at `https://ubergraph.apps.renci.org/sparql`
 - Not a website! Run a query at <https://api.triplydb.com/s/oQgG4f0gs>
- **Ontology Access Kit:** <https://github.com/INCATools/ontology-access-kit>
 - Comprehensive Python library with built-in connections to Ubergraph
 - OAK viz example:

```
runoak -i ubergraph: viz GO:0001750
-p rdfs:subClassOf,BFO:0000050
```
- **Downloads** of complete database available:
 - N-Quads file
 - Simplified node and edge tables



Ubergraph users

NCATS Biomedical Data Translator Program

- Developing a federated knowledge system capable of integrating existing biomedical data sets
- Will allow users to derive “insights that can accelerate translational research, support clinical care, and leverage clinical expertise to drive research innovations” <https://doi.org/10.1111/cts.12591>
- Many data sources in Translator reference OBO ontology terms
 - Ubergraph provides the contextual knowledge relating those terms to each other
- Ubergraph contains a graph of links to [Biolink Model](#) categories for OBO ontology classes; lingua franca for Data Translator

Ubergraph users

Mondo Disease Ontology development team

- Mondo is a large ontology with a complex graph structure referencing many external terms
- Ubergraph speeds creation of initial QC reports and exploration, which depend greatly on inferred connections
- After prototyping in Ubergraph, queries may be recoded to run against development files in the Mondo build pipeline

Ubergraph users

HuBMAP expert validation

- HuBMAP project is building a human reference atlas leveraging expert input
 - Focus on adult human anatomy
- Experts provide simple spreadsheets describing expected connections between Cell Ontology and Uberon anatomy terms (e.g., kidney cell types)
- Python-based workflow validates spreadsheets against inferred connections in Ubergraph, either confirming, or presenting alternatives
- Feedback loop:
 - HubMAP biologists map 'OFF-bipolar cell' (CL:0000750) to 'inner nuclear layer of retina' (UBERON:0001791),
 - This relation currently not found in Ubergraph via the Cell Ontology.
 - Validation tool searches for relationships among the terms in the HubMAP domain and finds that 'OFF-bipolar cell' currently does have a 'part_of' relationship with 'retina' (UBERON:0000966).
 - Editors consider adding an axiom to 'OFF-bipolar cell': 'part of' some 'inner nuclear layer of retina'.

Challenges

- **Logical incompatibilities among merged ontologies**
 - *Largely solved*: the majority of problems combining OBOs in the past were due to version mismatch rather than deep incompatibilities
 - Many ontologies now make available “base” release files, avoiding mixing various versions of imports
 - Ubergraph repo includes ontology pre-release integration tests as GitHub Actions
- **Unintended logical inferences** — “OWL is not modular”
 - Logical dependencies can result in new or changed inferences in a downstream ontology which may not yet have been vetted (e.g., GO and ChEBI example)
- **Scaling?**
 - The relation-graph expansion is about 6x the size of the input ontologies
 - So far scaling of the build has not presented a problem
 - Blazegraph query performance is frequently excellent, but possible to time out on particularly complex queries

Acknowledgments

Thank you:

- David Osumi-Sutherland, Anita Caron, Ugur Bayindir (EBI)
- Nico Matentzoglou (Semanticcy)
- Chris Mungall (LBNL)
- Chris Bizon (RENCI)
- Mac Chaffee (RENCI)

Renaissance Computing Institute: services hosting

Funding from NIH:

- Data Translator: 3OT2TR003449-01S1
- INCA tools: 5U01HG009453

Please join the #ubergraph channel on the OBO Slack!

Open an issue or discussion:

<https://github.com/INCATools/ubergraph>

