Developing Scientific Knowledge Graphs Using Whyis

James P. McCusker, Sabbir M. Rashid,
Nkechinyere Agu, Kristin P. Bennett, and Deborah L. McGuinness

Rensselaer Polytechnic Institute, Troy, NY 12180, USA

Abstract. We present Whyis, the first framework for creating custom provenance-driven knowledge graphs. Whyis knowledge graphs are based on nanopublications, which simplifies and standardizes the production of structured, provenance-supported knowledge in knowledge graphs. To demonstrate Whyis, we created BioKG, a probabilistic biology knowledge graph, and populated it with well-used drug and protein content from DrugBank, Uniprot, and OBO Foundry ontologies. As shown with BioKG, knowledge graph developers can use Whyis to configure custom knowledge curation pipelines using data importers and semantic extract, transform, and load scripts. Whyis also contains a knowledge meta-analysis capability for use in customizable graph exploration. The flexible, nanopublication-based architecture of Whyis lets knowledge graph developers integrate, extend, and publish knowledge from heterogeneous sources on the web.

1 Introduction

Successful scientific knowledge graph construction requires more than simply storing and serving graph-oriented data. Keeping a knowledge graph up to date can require developing a knowledge curation pipeline that either replaces the graph wholesale, whenever updates are made, or requires detailed tracking of knowledge provenance across multiple data sources. Additionally, non-deductive forms of knowledge inference have become important to knowledge graph construction. User interfaces are also key to the success of a knowledge graph-enabled application. Google's knowledge graph, for instance, takes the semantic type of the entity into account when rendering information about that entity. These challenges are dependent on high-quality knowledge provenance that, we claim, should be inherent in the design of any knowledge graph system, and not merely an afterthought. We present Whyis by creating and deploying an example knowledge graph aimed to support our previously custom-designed knowledge graph for drug repurposing [4]. BioKG, or Biological Knowledge Graph, uses data from DrugBank and Uniprot.[1] We also discuss the benefits of using our approach over using a conventional knowledge graph pipeline.

[1] http://drugbank.ca and http://uniprot.org, respectively.

2 Methods

To support these challenges, Whyis provides a semantic analysis ecosystem (Figure 1). This is an environment that supports research and development of semantic analytics for which we have previously had to build custom applications [4,5]. Users interact through views into the knowledge graph, driven by the type of node and view requested in the URL. Knowledge is curated into the graph through knowledge curation methods, including Semantic Extract, Transform, and Load (SETL), external linked data mapping, and Natural Language Processing (NLP). Autonomous inference agents in Whyis expand the available knowledge using traditional deductive reasoning as well as inductive methods that can include predictive models, statistical reasoners, and machine learning.

As a nanopublication-based knowledge graph, Whyis uses *nanopublications* to encapsulate every piece of knowledge introduced into the knowledge graphs it manages. Even in cases where the user does not explicitly provide provenance, Whyis collects provenance of who created the assertion and when. Introduced in [6] and expanded on in [2], a nanopublication is composed of three named RDF graphs: an *Assertion* graph, a *Provenance* graph, and a *Publication Info* graph. We see knowledge graphs that include the level of granularity supported by nanopublications as essential to fine-grained management of knowledge in knowledge graphs that are curated and inferred from diverse sources and can change on an ongoing basis. Other systems, like DBpedia[2] and Uniprot, have very rough grained management of knowledge using very large named graphs, which limit the ability to version, explain, infer from, and annotate the knowledge. With nanopublications, provenance of both the assertion and the artifact it is contained in is included in each edit of the knowledge graph. It is therefore possible to track user-contributed provenance as well as automatically collect provenance. For instance, Whyis tracks who composed the nanopublication and when it was edited using PROV-O and Dublin Core Terms. In the case of mapping in linked data, Whyis tracks where the graph was imported from. An example nanopublication from BioKG is shown in Figure 2. Whyis is written in Python using the Flask framework, and uses a number of existing infrastructure tools to work..

Knowledge Inference in Whyis is performed by a suite of "Agents", each performing the analogue to a single rule in traditional deductive inferencing. An agent is composed of a SPARQL query that serves as the "rule body" and a python function that serves as the "head". An agent is invoked when new nanopublications are added to the knowledge graph that match its SPARQL query. The rule body function produces RDF nanopublications as output, which are then in turn processed by other agents. The agent superclass assigns some basic provenance and publication information related to the given inference activity, which developers can add to in their implementations. Included inference agent types include entity extraction and resolution against existing knowledge graph

[2] http://dbpedia.org

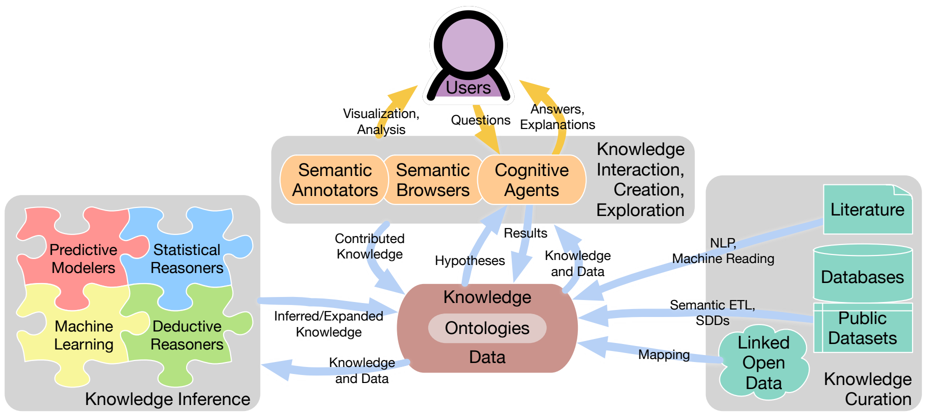


Fig. 1. The semantic ecosystem enabled by the Whyis framework for knowledge curation, interaction, and inference.

nodes, deductive reasoning agents that can be configured with custom rules, as well as many available pre-configured OWL 2 rules.

To create probabilistic knowledge graphs, Whyis implements a method of automated meta-analysis using the provenance of graph assertions when computing which nodes are related to each other. We refined the methods used in [4] by using Stouffer's Z-Method [8] to produce an overall probability for the link claim. We allow developers to determine, on a per-node type basis, what constitutes a link to or from that node. For biological entities in BioKG, outgoing links are defined as instances of *sio:Interacting*, where the current node is the *sio:hasAgent*, and the connected node, the *sio:hasTarget*. The default implementation uses *sio:ProbabilityMeasure* attributes on the nanopublication assertions. These attributes can be asserted directly, or developers can write an inference agent that computes values for them. If those attributes are not available, the default implementation looks at the number of references cited in the assertion's provenance, and assigns the configured base rate (in BioKG, focusing on results from published literature, we set it to 0.8) to each one. The resulting probabilities for each nanopublication that claims the link are then combined using Stouffer's Z-Method. The resulting graph is customizable to the domain; provides on-demand probabilities for each link; and is explorable, in that, for every known node in the graph, we are able to find related nodes. This neighborhood graph, as shown in Figure 3, is on the default page of every node, and has an interface that lets users search and explore the wider knowledge graph by expanding nodes, filtering on probability, and searching for nodes by name.

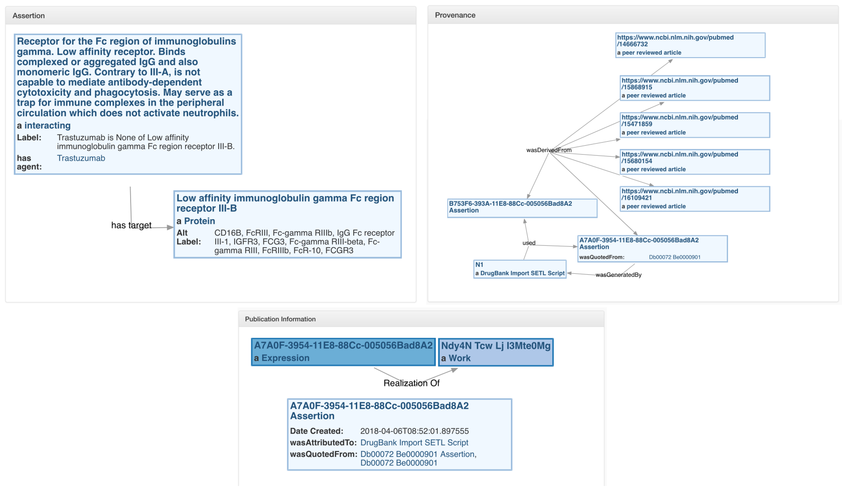

Assertion
Receptor for the Fc region of immunoglobulins gamma. Low affinity receptor. Binds complexed or aggregated IgG and also monomeric IgG. Contrary to III-A, is not capable to mediate antibody-dependent cytotoxicity and phagocytosis. May serve as a trap for immune complexes in the peripheral circulation which does not activate neutrophils.
a interacting
Label: Trastuzumab is None of Low affinity immunoglobulin gamma Fc region receptor III-B.
has agent: Trastuzumab
has target
Low affinity immunoglobulin gamma Fc region receptor III-B
a Protein
Alt Label: CD16B, FcRIII, Fc-gamma RIIIb, IgG Fc receptor III-1, IGFR3, FCG3, Fc-gamma RIII-beta, Fc-gamma RIII, FcRIIIb, FcR-10, FCGR3
Provenance
https://www.ncbi.nlm.nih.gov/pubmed/14666732
a peer reviewed article
https://www.ncbi.nlm.nih.gov/pubmed/15868915
a peer reviewed article
https://www.ncbi.nlm.nih.gov/pubmed/15471859
a peer reviewed article
https://www.ncbi.nlm.nih.gov/pubmed/15680154
a peer reviewed article
https://www.ncbi.nlm.nih.gov/pubmed/16109421
a peer reviewed article
wasDerivedFrom
B753F6-393A-11E8-88Cc-005056Bad8A2
Assertion
used
A7A0F-3954-11E8-88Cc-005056Bad8A2
Assertion
wasQuotedFrom: Db00072 Be0000901
N1
a DrugBank Import SETL Script
wasGeneratedBy
Publication Information
A7A0F-3954-11E8-88Cc-005056Bad8A2
a Expression
Ndy4N Tcw Lj I3Mte0Mg
a Work
Realization Of
A7A0F-3954-11E8-88Cc-005056Bad8A2
Assertion
Date Created: 2018-04-06T08:52:01.897555
wasAttributedTo: DrugBank Import SETL Script
wasQuotedFrom: Db00072 Be0000901 Assertion, Db00072 Be0000901

Fig. 2. An example nanopublication from BioKG. Scientific knowledge is asserted in the Assertion graph, while justification of that knowledge (that it is supported by a paper) is placed in the Provenance graph. Information about how the nanopublication was generated is placed in the Publication Information graph. If the Semantic ETL script is modified, since this nanopublication was generated from it, the old version is removed and a new one computed. This process applies to inference agent outputs as well.

3 Related Work

Some existing frameworks support some of Stardog[3] includes OWL reasoning, mapping of data silos into RDF, and custom rules. Ontowiki provides a user interface on top of an RDF database that tracks history, allows users to browse and edit knowledge, and supports user interface extensions [4]. Callimachus lets developers create UIs by object type using RDFa [1]. Virtuoso Openlink Data Spaces is a linked data publishing tool that provides a set of pre-defined data import tools and a fixed set of views on the linked data it creates.[5] Vitro[6] supports the creation of new ontology classes and instances, but does not allow users to create custom interfaces.

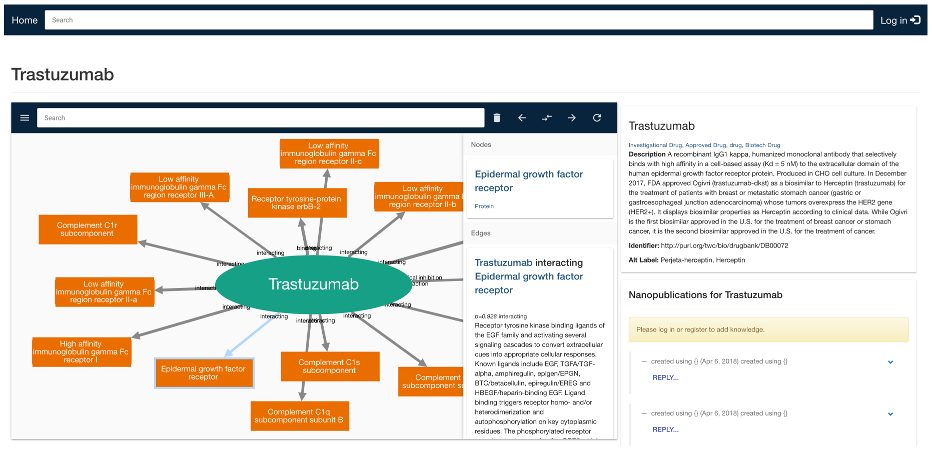
4 Building BioKG using Whyis

We show Whyis by building a biology knowledge graph, BioKG, at http://bit.ly/whyis-demo. BioKG, uses the built-in knowledge exploration and semantic meta analysis capabilities of Whyis to support probabilistic knowledge

[3] A case study: https://www.stardog.com/blog/nasas-knowledge-graph/
[4] http://ontowiki.net
[5] https://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/Ods
[6] Available: https://github.com/vivo-project/Vitro

graph exploration. Whyis supports a large number of knowledge graph use cases that supported the development of BioKG in areas of knowledge curation, interaction, and inference, which we have detailed on the Whyis website.[7] The code used to configure BioKG is available on GitHub at https://github.com/tetherless-world/biokg. BioKG provides on-demand mappings to entities in OBO Foundries[8], Uniprot, DOI [7], and DBpedia, and has SETL scripts for loading DrugBank. Users can explore the graph through search and the list of recently updated entities on the default landing page.

The BioKG knowledge graph was written in 3 active development days using 614 lines of code, including templates, configuration, and data curation scripts. The conversion of DrugBank to RDF is especially notable. In the original drug repurposing project, we used an Extensible Stylesheets Language Template (XSLT) to generate RDF. Many conversion errors were introduced through the need to generate RDF as plain text, including occasional spaces in URIs and other invalid syntax. The SETL script, on the other hand, was able to validate the RDF as it was generated, and the resulting template is much more legible.


Home
Search
Log in
Search
Low affinity immunoglobulin gamma Fc region receptor II-c
Low affinity immunoglobulin gamma Fc region receptor III-A
Receptor tyrosine-protein kinase erbB-2
Low affinity immunoglobulin gamma Fc region receptor II-b
Complement C1r subcomponent
Trastuzumab
Low affinity immunoglobulin gamma Fc region receptor II-a
High affinity immunoglobulin gamma Fc receptor I
Epidermal growth factor receptor
Complement C1s subcomponent
Complement C1q subcomponent subunit B
interacting
Nodes
Edges
REPLY...

Trastuzumab

Epidermal growth factor receptor
Protein

Trastuzumab interacting Epidermal growth factor receptor

p=0.928 interacting
Receptor tyrosine kinase binding ligands of the EGF family and activating several signaling cascades to convert extracellular cues into appropriate cellular responses. Known ligands include EGF, TGFA/TGF-alpha, amphiregulin, epigen/EPGN, BTC/betacellulin, epiregulin/EREG and HBEGF/heparin-binding EGF. Ligand binding triggers receptor homo- and/or heterodimerization and autophosphorylation on key cytoplasmic residues. The phosphorylated receptor

Trastuzumab

Investigational Drug, Approved Drug, drug, Biotech Drug
Description A recombinant IgG1 kappa, humanized monoclonal antibody that selectively binds with high affinity in a cell-based assay (Kd = 5 nM) to the extracellular domain of the human epidermal growth factor receptor protein. Produced in CHO cell culture. In December 2017, FDA approved Ogivri (trastuzumab-dkst) as a biosimilar to Herceptin (trastuzumab) for the treatment of patients with breast or metastatic stomach cancer (gastric or gastroesophageal junction adenocarcinoma) whose tumors overexpress the HER2 gene (HER2+). It displays biosimilar properties as Herceptin according to clinical data. While Ogivri is the first biosimilar approved in the U.S. for the treatment of breast cancer or stomach cancer, it is the second biosimilar approved in the U.S. for the treatment of cancer.

Identifier: http://purl.org/twc/bio/drugbank/DB00072

Alt Label: Perjeta-herceptin, Herceptin

Nanopublications for Trastuzumab

Please log in or register to add knowledge.

— created using {} (Apr 6, 2018) created using {}

— created using {} (Apr 6, 2018) created using {}

Fig. 3. A knowledge graph node page, showing related graph nodes with edges weighted by their probability as computed by automated meta-analysis.

4.1 Future Work

We continue to develop the content, user interface, and inference capabilities of Whyis and BioKG to research beyond the probabilistic graph exploration laid out in [4]. We are developing new methods for display in the knowledge explorer user interface. We are working on a suite of integrated inference agents. Inductive

[7] http://tetherless-world.github.io/whyis/usecases
[8] http://obofoundry.org

agents will use statistics and machine learning algorithms to infer new knowledge and relationships in the knowledge graph. For example, we plan to investigate graph learning algorithms like [3] that can learn from the structure of published knowledge graphs to find new relations.

5 Conclusions

We introduced Whyis as the first provenance-aware open source framework for knowledge graph development that supports knowledge graph curation, interaction, and inference within a unified semantic analysis ecosystem. We discussed the importance of nanopublications in the architecture of Whyis and why it is valuable to develop knowledge graphs that are built on nanopublications. We showed how Whyis is able to create the biological knowledge graph BioKG, and how we use semantic meta-analysis to support its use as a probabilistic knowledge graph. Whyis is published under the Apache 2.0 License on Github[9], with additional documentation for how to develop custom knowledge graphs.

Acknowledgements

This work was funded by NIEHS Award 0255-0236-4609 / 1U2CES026555-01, NSF Award OAC-1640840 IBM Research AI Horizons Network, and by the Gates Foundation through HBGDki.

References

1. Battle, S., Wood, D., Leigh, J., Ruth, L.: The callimachus project: Rdfa as a web template language. In: Proceedings of the Third International Conference on Consuming Linked Data-Volume 905. pp. 1–14. CEUR-WS. org (2012)
2. Groth, P., Gibson, A., Velterop, J.: The anatomy of a nanopublication. Information Services and Use 30(1), 51–56 (2010)
3. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: AAAI. vol. 15, pp. 2181–2187 (2015)
4. McCusker, J.P., Dumontier, M., Yan, R., He, S., Dordick, J.S., McGuinness, D.L.: Finding melanoma drugs through a probabilistic knowledge graph. PeerJ Computer Science 3, e106 (Feb 2017), https://doi.org/10.7717/peerj-cs.106
5. McGuinness, D.L., Bennett, K.: Integrating semantics and numerics: Case study on enhancing genomic and disease data using linked data technologies. Proceedings of SmartData pp. 18–20 (2015)
6. Mons, B., Velterop, J.: Nano-publication in the e-science era. In: Workshop on Semantic Web Applications in Scientific Discourse (SWASD 2009). pp. 14–15 (2009)
7. Paskin, N.: Digital object identifier (doi®) system. Encyclopedia of library and information sciences 3, 1586–1592 (2010)
8. Whitlock, M.C.: Combining probability from independent tests: the weighted z-method is superior to fisher's approach. Journal of Evolutionary Biology 18(5), 1368–1373

[9] https://tetherless-world.github.io/whyis