Transforming and Unifying Research with Biomedical Ontologies

The Penn TURBO project

Christian J. Stoeckert Jr.
Dept. of Genetics, Institute for Biomedical Informatics
Perelman School of Medicine, University of Pennsylvania
Philadelphia, PA, USA
stoeckrt@pennmedicine.upenn.edu

David Birtwell, Heather Williams
Penn Medicine BioBank, Institute for Translational Medicine and Therapeutics, Perelman School of Medicine, University of Pennsylvania
Philadelphia, PA, USA

Hayden Freedman, Mark A. Miller
Institute for Biomedical Informatics
Perelman School of Medicine, University of Pennsylvania
Philadelphia, PA, USA

***Abstract*— The Penn TURBO (Transforming and Unifying Research with Biomedical Ontologies) project aims to accelerate finding and connecting key information from clinical records for research through semantic associations to the processes that generated the clinical data. Major challenges to using clinical data for research are integrating data from different sources which may contain multiple references to the same entity (e.g., person, health care encounter) and incomplete or conflicting information (e.g., gender, BMI). There is also the need to track the provenance of information used when making decisions on what is the actual phenotype of a person. We take a realism-based ontology approach to address these problems through transformation and instantiation of clinical data with an OBO-Foundry based application ontology in a semantic graph database. We have developed an application stack and used it on an 11,237 whole exome sequencing patient cohort capturing key demographics, diagnosis codes, and prescribed medications. The anticipated payoff is to be able to make use of inferencing provided by the semantics to classify and search for instances of people and specimens with desired characteristics.**

Keywords—realism-based ontology; OBO Foundry; referent tracking; clinical data; diagnosis codes; prescriptions

I. Introduction

The goal of the TURBO project is to transform and unify research data with biomedical ontologies. Typically data are obtained in tabular form often from relational databases. The column headers and row values are often idiosyncratic and even when based on a standard may be malformed, incomplete, and contradictory. Dependencies and deep relations between the headers (data variables) and values are rarely explicit. Transforming the data into a semantic graph instantiating a realism-based ontology allows us to state what is known about people and what has happened to them, what information is available about them, and what conclusions can be drawn based on that information. Clinical data often comes from multiple sources (e.g., EPIC, REDCap). Instantiation of data from different sources in the same realism-based ontology [1] allows us to unify the data. Part of the unification comes through referent tracking [2], associating information for the same person, quality, or event with a unique identifier for that referent regardless of where and when the information was obtained.

The Open Biomedical Ontologies Foundry [3] provides through its library of ontologies the ability to create a biomedical ontology that is realism-based. We created the TURBO ontology as an application ontology based on these ontologies drawing from the Ontology for Biomedical Investigations (OBI) [4] and the Ontology for Biobanking (OBIB) [5] in particular. By application ontology, we mean that we are primarily reusing terms (classes, instances, and relations) from existing ontologies and creating terms only as needed to move the project forward. Terms that potentially have broader usage are submitted to existing ontologies.

An application stack called Drivetrain was developed to perform part of the transformation, the unification, referent tracking, and generating conclusions as RDF statements about people and their qualities. Currently the Karma tool [6] is used to transform tabular data into initial RDF triples for Drivetrain to use. Ontology modeling is also used to capture provenance of data and conclusions drawn based on the data. After running the Drivetrain stack, the reasoning capabilities of the semantic graph database can be used to classify and aid search for instances of people and specimens with desired characteristics. For example, people can be identified who have been prescribed a particular class of drugs ('statins'). We intend to create phenotypic profiles in the form of equivalence axioms that will be used to infer which people or specimens match those profiles.

II. Methods

A. Technologies used in TURBO

Ontotext GraphDB (version 8.4.1) [7] is the semantic graph database used. Scala (version 2.11) [8] is used for programmatic interaction with the database, leveraging the RDF4J (version 2.2.2) library [9]. UUIDs are generated using the randomUUID() method found in the java.util.UUID

TURBO is supported by the Institute for Biomedical Informatics and the Institute for Translational Medicine And Therapeutics at the University of Pennsylvania Perelman School of Medicine.

package [10]. LIBSVM was used through the svm() function from R e1071 [11].

The TURBO ontology was generated following the approach described in [12]. Terms were selected from OBIB using Ontodog [13] and additional terms were imported using the OntoFox tool [14]. New terms were added using Protégé [15].

B. TURBO content

Data on a whole exome sequencing cohort of 11,237 participants ('biobank consenters') have been used to populate a GraphDB database. The data include information on gender identity, date of birth, and body mass index (BMI, calculated from height and weight) collected during 14,450 biobank encounters and 98,585 health care encounters. In addition, 181,420 diagnosis codes and 136,249 medications were obtained during health care encounters. The data was obtained from relational tables provided by the Penn Medicine Biobank from two sources, a data warehouse and REDCap.

In addition to RDF triples generated from the data, individual ontologies and terminologies were also loaded into the GraphDB database. The ontologies included the TURBO application ontology, RDF representations of ICD9 and ICD10 codes obtained from the NCBO Bioportal [16], all portions of the Drug Ontology [17] except NDC annotation, the "lite" component of ChEBI [18], and the Monarch Disease Ontology (MonDO) [19].

C. Generation of RDF triples to load into the TURBO GraphDB database.

The Karma application (version 2.1) was used to generate RDF triples from the tabular data for loading into the GraphDB database. Karma models were based on the TURBO ontology.

D. TURBO code and documentation

The code base for the Drivetrain component is available at GitHub including documentation of the full TURBO stack and description of ontology modeling. https://pennturbo.github.io/Turbo-Documentation/

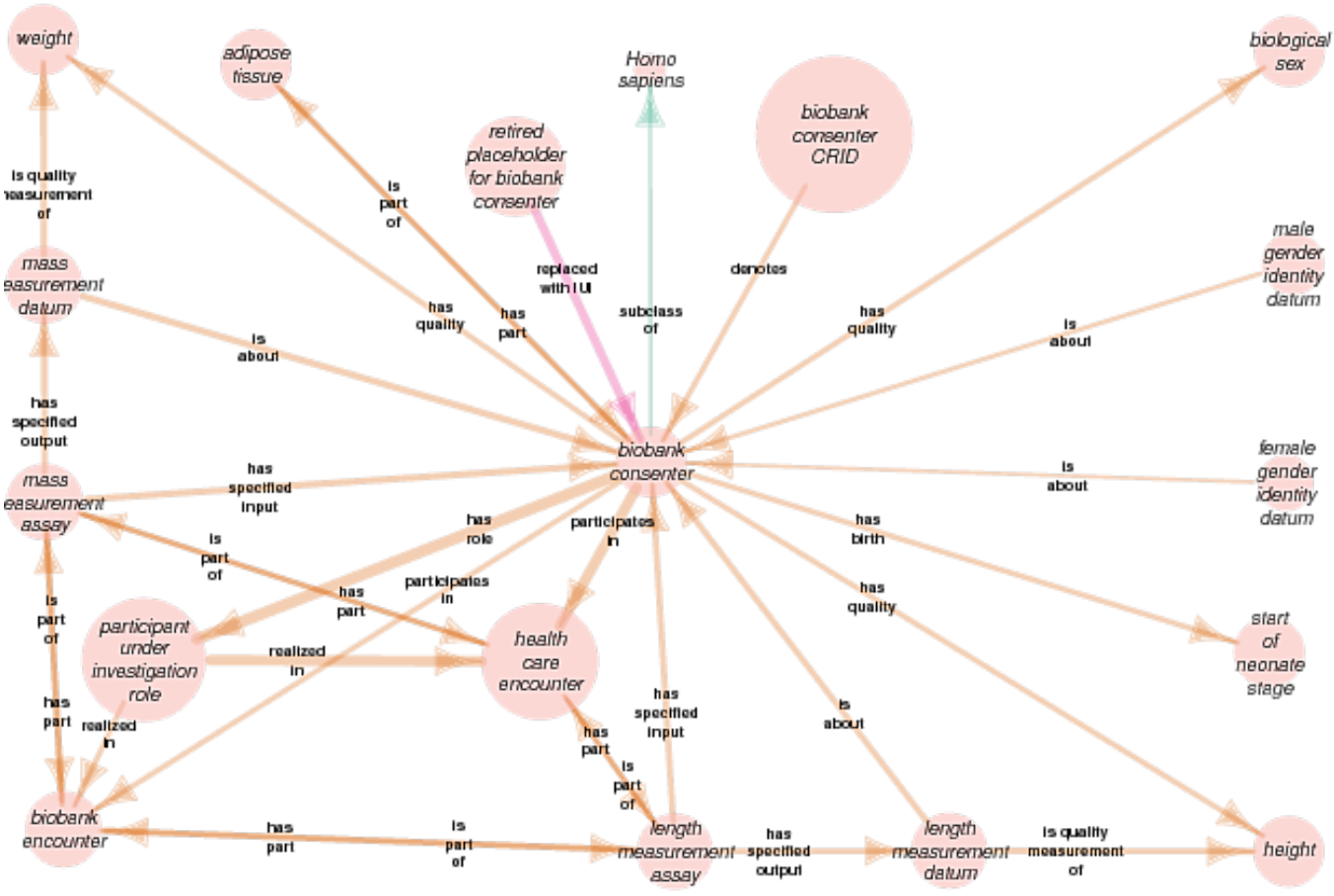

Figure 1. A graph depicting instantiated parts of the TURBO ontology including 'biobank consenter'. Nodes are classes whose size reflects usage in the instantiation of the WES cohort data. Edges are object properties (including the green 'subclass of' but with the exception of the pink edge) whose width also indicates usage. The one exception is a pink annotation property indicating that a 'retired placeholder for biobank consenter' was 'replaced with' 'biobank consenter' as a result of the referent tracking process.

III. Results

A technology stack has been developed for the TURBO project that implements a pipeline to transform tabular data into semantic triples, stored in a Resource Description Framework (RDF) triple store, using terms from the TURBO Ontology (https://raw.githubusercontent.com/PennTURBO/Turbo-Ontology/master/ontologies/turbo_merged.owl). The TURBO ontology at time of writing consists of 727 terms (415 classes, 41 individuals, 271 properties). These are primarily drawn from 25 ontologies with 161 new terms created for TURBO (69 classes, 19 individuals, 73 properties). URIs and all labels of terms instantiated in the current TURBO semantic repository are listed at the bottom of: https://pennturbo.github.io/Turbo-Documentation/turbo-ontology.html (along with a discussion and an example of an instantiated triple higher on the page). Terms in the TURBO ontology are focused on patients and their qualities along with information collected on them, 'health care encounter's (http://purl.obolibrary.org/obo/OGMS_0000097) and their outputs (diagnoses, measurements), and biobank encounters and their outputs. The new terms mainly cover shortcut relations utilized in the Karma mapping and for managing UUIDs during referent tracking. At the Penn Medicine Biobank, data are collected when participants are consented at which time they have not yet donated a specimen but have been assigned an ID. To capture this case, a 'biobank consenter' term has been generated defined as a participant in a biobank consenting process (Figure1). Incorporating the essence of this term is in progress with ICO [20] and OBIB developers.

The Karma tool was used to map relational data to ontology terms saved with an extended version of the R2RML language. The mappings were then used to publish the data as RDF triples. The initial RDF triples make use of shortcut relation properties to simplify the manual mapping. The essence of TURBO shortcut relations is to allow a minimal number of classes to be instantiated – frequently just one. For example, an input table nominally about health care encounters may include height, weight and body mass index (BMI) values. Those data items are not values of the encounters, but rather values of properties borne by the people who participated in the encounters. The shortcut relation "shortcut health care encounter to BMI" eliminates the need to instantiate a class that represents the encounter participants and instead says that there is some path from the encounter to the BMI value. The Drivetrain application (described next) contains all of the logic necessary to expand the shortcut into a semantically complete description of reality.

The Drivetrain application was built to load and process the RDF triples with the following steps:

A. *Shortcut RDF Triples and TURBO ontology loaded to an Ontotext GraphDB repository*

During the data import step, the input data are written to an isolated section of the graph. The triples are not expected to have globally unique identifiers and so must be sectioned off from all other data in the triple store.

B. *EXPAND Queries create fully ontologized model from shortcut triples*

The shortcut expansion phase takes all triples in the input data that use shortcut relations and expands them to fully ontologized forms. A single shortcut triple will likely expand to multiple ontologized triples. In addition to expanding the triples, the Internationalized Resource Identifiers (IRIs) in the imported data are made unique using Universally Unique Identifiers (UUIDs). After this phase is complete, the data in the isolated import graph have globally unique identifiers and are fully ontologized, though they may not yet be ready to be incorporated into the rest of the triple store.

Data integrity rules are applied to all triples in the isolated import graph to assure that the data meet the minimum level of integrity required by the Drivetrain application. Several conditions must be met to pass, including checks that all classes and properties present in the incoming data must also be present in the TURBO ontology, all denoted registries must be represented in the ontology, and all dates must be parseable, reasonable, and be typed as dates. If all integrity checks have passed, then the data are ready to be connected to the rest of the graph.

C. *Scala-based REFERENT TRACKER combines duplicate entities*

During the Referent Tracking phase, all instantiated IRI-bearing terms that singularly and uniquely refer to a single thing in reality are replaced with a single Instance Unique Identifier (IUI), which is implemented by Drivetrain as an IRI that specifically contains a Universally Unique Identifier value (UUID). After this phase is complete, the RDF data are normalized such that all entities in reality can be identified by a single unique identifier that is independent yet connected to the source relational data (Figure 2).

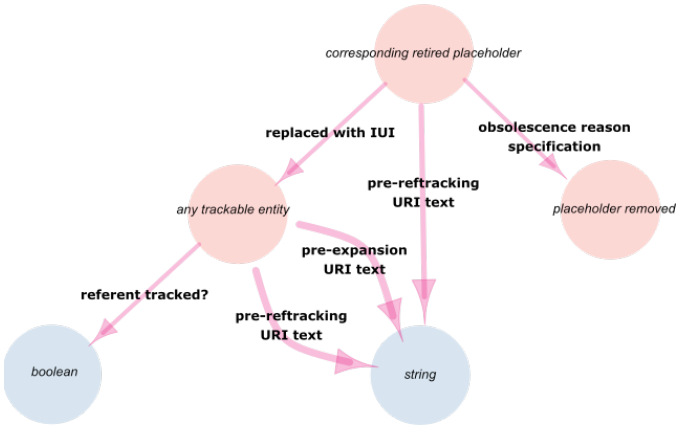

Figure 2. Prototypical referent tracking. Blue nodes are literals. Edges are annotation properties providing provenance for referent tracking.

Since our data comes from many sources, it is possible that the same 'biobank consenter' may appear in multiple data sources, each of which may contain different or contradicting

information. It is the goal of the Referent Tracker to apply custom rules in order to determine when two consenters must be combined into one. Likewise, the same encounter may also appear in multiple data sources. A simple rule is that the identifier and identifier source (central registry ID symbol and registry) associated with the entity are the same.

D. Scala-based ENTITY LINKER links Health care and Biobank Encounters to Biobank Consenters

Entity Linking is a generic term used here to mean the process of attaching consenters to their encounters based on data provided by a relational Join table. This process is necessary because consenters and their encounters may be received in separate files. Drivetrain can make matches by comparing the literal values of encounter symbols and consenter symbols, and the values of the respective registries.

E. Scala-based CONCLUSIONATOR creates inferences about Dates of Birth, Biological Sex, and BMI

During the conclusionating phase, rules are applied to the data to generate statements about a person or event. Currently this is done to resolve potentially conflicting data to single conclusions, which can be used for querying purposes. The potentially conflicting data derived from the sources remain in the graph and can be queried. In the future, it will be used to combine data of different types (e.g., diagnosis code, medication, lab test result) to make a single statement (e.g., a person is diabetic). To facilitate easy querying, the conclusions, which are RDF triples, are placed in a separate named graph. After this phase is complete, there will be a named graph of conclusions, which contains simplified non-conflicting statements. Conclusionating is applied to generate statements about the consenter's biological sex, date of birth, and BMI at the date of each biobank encounter. The rules used for drawing conclusions are currently very simple, but the system is envisioned to handle more complex rules and be able to draw on a library of different rules in the future.

One way to calculate BMI is by performing a computation over a person's height and weight, which can be measured during a health care encounter or recorded on a case report form during study recruitment during a biobank encounter (when a person becomes a 'biobank consenter'). It is useful to know the BMI of biobank consenters at their date of recruitment.

It is not guaranteed that the source data required to calculate BMI at date of biobank encounter will be both available and of sufficient quality. It may be that height and weight measurements were recorded at the health care encounter, the biobank encounter, neither, or both. Further, the data may have been recorded improperly, which would result in a calculated BMI that is outside the acceptable range.

The following rules are currently applied to account for these situations:

For each date of recruitment for each person:

- If there are in-range height and weight measurements recorded in the health care encounter on the date of recruitment, compute the BMI and conclude that it is the person's BMI at the given date of recruitment.
- If the BMI cannot be computed from the health care encounter, but there are valid height and weight measurements records on the case report form filled out as part of the study recruitment process, compute the BMI from the case report form data and conclude that it is the person's BMI at the given date of recruitment.
- If neither the health care encounter nor the study recruitment encounter yield a BMI conclusion, then record that BMI for this given date of recruitment is inconclusive.

F. Diagnosis Data is mapped by cross-referencing ICD9/ICD10 hierarchies and MonDO ontologies

Diagnosis codes come to TURBO in the form of ICD9 and ICD10 codes [21]. In order to enable searches broader than a single code value, we load RDF versions of ICD9 and ICD10 downloaded from the NCBO Bioportal, which provide subClassOf relations. We also load MonDO, an aggregation of disease ontologies including the Human Disease Ontology [22]), which includes database cross references for ICD codes. We use these cross references to create mentions between diagnosis codes and diseases, thereby enabling disease-based searches.

G. Medication Order Name Data are mapped to ontologies using Solr indexed text search and a Support Vector Machine (SVM)

Medication orders are provided primarily as free text, often including dosage and route of administration information. Associating these orders to terms in ChEBI (Chemical Entities of Biological Interest) would enable searches based on the parent classes of active ingredients and their roles. To accomplish this, the orders are computationally mapped to terms from the Drug Ontology (DRON) which provides cross-references to ChEBI. About 30% of the distinct medications prescribed to our WES cohort also came with RxNorm identifiers [23] that could be directly associated to DRON and ChEBI via direct cross references. The RxNorm associations were then used as a training set for machine learning (LIBSVM) using results from the string matching output from Apache Solr [24]. For the WES cohort, we were able to map 86.1% of distinct medications (sensitivity = 0.98; specificity = 0.95) covering 88% of the total medications prescribed (excluding non-drug prescriptions).

H. Performance

The complete Drivetrain stack was run on a linux application server with 8 GB RAM and 2 processors and a GraphDB database server with 64 GB RAM and 4 processors.

The run from loading of graph through medication mapping (steps described in sections A through G above) took 82 minutes for the WES cohort data and supportive ontologies. It resulted in 25,521,235 triples. About 3.6 million triples were initially loaded and then expanded to about 12 million triples. Additional triples resulted from referent tracking, conclusionating, and adding diagnosis and medication terms and associations.

Searches for diagnosis classes take approximately a second. For example, a search for all participants in a health care

encounter which resulted in a diagnosis that mentions 'myocardial infarction' will return those assigned a ICD10 code of I21.3 (acute myocardial infarction).

Searches for medications also take on the order of seconds. A search for all participant prescribed a 'statin' returned all appropriate statins and no inappropriate ones based on drug name matches and their active ingredients with one important exception. Crestor contains rosuvastatin but is not identified as a statin. That is because rosuvastatin while present in both DRON and ChEBI have different IRIs. We are able to address this issue locally by using equivalence statements between the two (we are also following up with DRON to resolve this issue).

IV. Discussion

The TURBO project is currently in active development as a demonstration project for the Penn Institute for Biomedical Informatics. We have a stable application stack, Drivetrain, that combined with the Karma tool, enabled us to transform, load, referent track, and make conclusions related to a real dataset of interest, a WES cohort of 11,237 participants. Unlike traditional data warehousing, the TURBO system performs integration through rules applied during referent tracking and conclusionating. The processes used to determine when entities are the same (people, encounters) in referent tracking or make statements about a person (e.g., BMI) in conclusionating are modeled in the ontology and stored in the graph for provenance. Thus, Drivetrain provides an ontology-supported knowledge layer along with the loaded data.

User stories, common requests by researchers searching clinical data, are driving TURBO development. Competency questions based on these user stories are then used to evaluate the system. Examples include identification of people of specified age, biological sex, and BMI. These are possible as is finding those who have been prescribed a particular class of drugs and assigned a diagnosis code linked to a particular class of disease. We are currently working on adding genotype data resulting from exome sequencing. Future additions will include laboratory tests.

Scalability of the system remains to be determined. We plan to expand both the number of participants and type of data instantiated in the semantic graph database. At 25 million triples, our current graph database has room to grow. We run Drivetrain with reasoning off but can then load into a graph database with RDFS+ or OWL-Horst reasoning turned on. For the current datasets this takes less than an hour. We are also exploring loading shortcut triples generated by alternative methods to Karma that are less manual.

Our efforts at medication mapping have used standard tools with good success but we would like to improve coverage as much as possible. Some prescriptions are not medications at all (e.g., wheelchairs, saline solutions, etc.) and we can generate lists to recognize these. We will explore use of other terminologies (e.g., NDFRT [25]) that may provide routes through active ingredients and equivalence matches to entries in ChEBI. Once we have a ChEBI IRI linked to a prescription it then can be searched based on the structure or role of the active ingredient.

The TURBO project represents a new direction in applying ontologies to clinical data. Most efforts do not explicitly involve realism-based ontologies or if they do use them it is in the form of associations and not instantiations. However, there are related projects instantiating OBO and realism-based ontologies. These include ones by William Duncan (Roswell Park) [26], by Amanda Hicks and William Hogan (U. Florida) [27], and by Bjoern Peters (LaJolla Institute for Immunology) [28] although they don't do referent tracking or conclusionating as in TURBO. This growing number of independent efforts raise the exciting potential of linking such systems together.

Ultimately, we intend for the TURBO project to provide a Phenotype Storefront that users can query to find participants and specimens of interest. The current plan is to just return the number of hits as results and require IRB approval for accessing identifiable data. We also want to learn from searches made by investigators in order to generate defined classes of participants and specimens. For example, equivalence axioms for someone who has had a particular disease course could include an appropriate diagnosis code but also a relevant prescription and laboratory test result. Inferencing applications of this nature will bring to bear the power of ontologies to provide what can't be done by traditional relational systems.

Acknowledgment

All the authors have been approved under IRB protocol 813913 from the University of Pennsylvania to work with the described patient data. We thank Werner Ceusters and William Hogan for their advice and feedback on implementation of referent tracking. We also thank Jason Moore, Scott Damrauer, Michael Feldman, Peter Gabriel, John Holmes, and Daniel Rader for their support and guidance as the TURBO governance board.

References

[1] B. Smith and W. Ceusters, "Ontological realism: A methodology for coordinated evolution of scientific ontologies," Appl Ontol. 2010 Nov 15;5(3-4):139-188.

[2] W. Ceusters and B. Smith, "Strategies for referent tracking in electronic health records," J Biomed Inform. 2006 Jun;39(3):362-78.

[3] B. Smith, et al., "The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration," Nat Biotechnol, 2007. 25(11): p. 1251-5.

[4] A. Bandrowski, et al., "The Ontology for Biomedical Investigations," PLoS One, 2016. 11(4): p. e0154556.

[5] M. Brochhausen, et al., "OBIB-a novel ontology for biobanking," J Biomed Semantics, 2016. 7: p. 23.

[6] C. A. Knoblock, et al., "Semi-Automatically Mapping Structured Sources into the Semantic Web," ESWC'2012

[7] Ontotext GraphDB. https://ontotext.com/products/graphdb/

[8] The Scala Programming Language. https://www.scala-lang.org/

[9] Eclipse RDF4J. http://rdf4j.org/

[10] Class UUID. https://docs.oracle.com/javase/8/docs/api/java/util/UUID.html

[11] R e1701e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. https://cran.r-project.org/web/packages/e1071/index.html

[12] J. Zheng, E. Manduchi, and C. J. Stoeckert, "Development of an application ontology for beta cell genomics based on the ontology for biomedical investigations," CEUR Workshop Proceedings, 1060, 62-67, 2013.

[13] J. Zheng, Z. Xiang, C. J. Stoeckert Jr., and Y. He, "Ontodog: a web-based ontology community view generation tool" Bioinformatics, 2014 May 1;30(9):1340-2.

[14] Z. Xiang, M. Courtot, R. R. Brinkman, A. Ruttenberg, and Y. He, "OntoFox: web-based support for ontology reuse," BMC Res Notes. 2010 Jun 22;3:175.

[15] M. A. Musen, "The Protégé project: A look back and a look forward. AI Matters," Association of Computing Machinery Specific Interest Group in Artificial Intelligence, 1(4), June 2015.

[16] P. L. Whetzel, et al., "BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications" Nucleic Acids Res. 2011 Jul;39(Web Server issue):W541-5.

[17] W. R. Hogan, et al., "Therapeutic indications and other use-case-driven updates in the drug ontology: anti-malarials, anti-hypertensives, opioid analgesics, and a large term request," J Biomed Semantics. 2017 Mar 3;8(1):10.

[18] J. Hastings, et al., "ChEBI in 2016: Improved services and an expanding collection of metabolites," Nucleic Acids Res. 2016 Jan 4;44(D1):D1214-9

[19] Monarch Disease Ontology. http://obofoundry.org/ontology/mondo.html

[20] Inormed Consent Ontology (ICO). https://github.com/ICO-ontology/ICO

[21] World Health Organization International Classification of Diseases. http://www.who.int/classifications/icd/en/

[22] W. A. Kibbe, et al., "Disease Ontology 2015 update: an expanded and updated database of human diseases for linking biomedical knowledge through disease data," Nucleic Acids Res. 2015 Jan;43(Database issue):D1071-8.

[23] RxNorm. https://www.nlm.nih.gov/research/umls/rxnorm/

[24] Apache Solr. http://lucene.apache.org/solr/

[25] NDFRT (National Drug File - Reference Terminology) – Synopsis. https://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/NDFRT/

[26] T. Thyvalikakath, et al., "National Dental PBRN. Restorative/Endodontic Procedures Performed in National Dental PBRN Practices," J Dent Res 97 (Spec Iss): 2859794, 2018.

[27] PCORowl. https://zenodo.org/record/1241209#.WvoBFsgh2L4

[28] R. Vita, J. A. Overton, J. A. Greenbaum, A. Sette, OBI consortium, and B. Peters, "Query enhancement through the practical application of ontology: the IEDB and OBI," Journal of Biomedical Semantics20134(Suppl 1):S6.