queryMed package: how to annotate medicine and pathology codes for pharmaco-epidemiological studies

Y. Rivault, O.Dameron, and N. Le Meur

03/08/2018

Introduction :

Because medical data, for example drugs and diseases, is often codified according to international nomenclatures, it can be linked to knowledge representations from medical and pharmacological domains. This can help improving data analysis by enriching the information it contains, for example by mining drug-drug interactions in a database of drug consumption (Pathak, Kiefer, and Chute 2013).

Semantic Web technologies and Linked Data initiatives have led to the spread of knowledge representations through ontologies, thesauri, taxonomies and nomenclatures. By providing standards and technologies for knowledge representation, integration and interrogation, the Semantic Web supports both technical and semantic interoperability for knowledge sharing and reuse. But if several Linked Data initiatives have published medical and pharmacological ontologies, the use of these standards, technologies and knowledge representations is still hesitant by the statisticians who deal with healthcare data (Ferreira et al. 2013).

The queryMed package purpose is to provide a user-friendly way to accesss the main medical and pharmacological knowledge sources from the Linked Data, through R, and linking them to healthcare data, so that the biostatisticians, epidemiologist and pharmaco-epidemiologists could enrich the data they analyze.

Installing queryMed

To retrieve and install queryMed, for the first time through github, you can use *devtools* R package:

```
install.packages("devtools")
devtools::install_github("yannrivault/queryMed/queryMed")
```

To load queryMed call the *library()* function:

```
library(queryMed)
```

SPARQL

SPARQL is one of the standards from the Semantic Web. It allows to query knowledge and data written in the Semantic Web representation standards (e.g. RDF and OWL). Some remote servers, called SPARQL endpoints, give access to such data and knowledge. As you might have already guessed, it can be queried with SPARQL. There are many SPARQL endpoints that are fully or partly dedicated to biomedical knowledge : BioPortal(Salvadores et al. 2013), Bio2rdf (Callahan et al. 2013), Ontobee (Ong et al. 2017) or also DB-pedia (Lehmann et al. 2015).

queryMed offers an elementary function to send SPARQL queries over SPARQL endpoints from the Web.

Here is an example of a SPARQL query, sent on bio2rdf :

```
query=
"SELECT DISTINCT *
  WHERE  {
    ?db <http://bio2rdf.org/drugbank_vocabulary:x-atc> ?atc .
```

```
    ?db dcterms:title ?title .
    ?db rdfs:label ?label .
    ?db dcterms:description ?description .
    ?db <http://bio2rdf.org/drugbank_vocabulary:category> ?category .
  }
limit 5
"

res=sparql(query,url="http://bio2rdf.org/sparql")
```

```
## Querying http://bio2rdf.org/sparql
```

```
res
```

```
## # A tibble: 5 x 6
##   db      atc     title label  description                      category
##   <chr>   <chr>   <chr> <chr>  <chr>                            <chr>
## 1 http:/~ http:/~ Algl~ Alglu~ Human Beta-glucocerebrosidase o~ http://bi~
## 2 http:/~ http:/~ Laro~ Laron~ Human recombinant alpha-L-iduro~ http://bi~
## 3 http:/~ http:/~ Cycl~ Cyclo~ A cyclic undecapeptide from an ~ http://bi~
## 4 http:/~ http:/~ Cycl~ Cyclo~ A cyclic undecapeptide from an ~ http://bi~
## 5 http:/~ http:/~ Cycl~ Cyclo~ A cyclic undecapeptide from an ~ http://bi~
```

If Uniform Resource Identifier (URI) is a standard in the Semantic Web, it is not so convenient from a statistician point of vue. Let's turn it into normal data with *uri2norm()*.

```
uri2norm(res)
```

```
## # A tibble: 5 x 6
##   db      atc     title        label  description                 category
##   <chr>   <chr>   <chr>        <chr>  <chr>                       <chr>
## 1 DB00088 A16AB01 Alglucerase  Alglu~ Human Beta-glucocerebrosid~ Enzyme--~
## 2 DB00090 A16AB05 Laronidase   Laron~ Human recombinant alpha-L--~ Enzyme--~
## 3 DB00091 L04AD01 Cyclosporine Cyclo~ A cyclic undecapeptide fro~ Antifun~
## 4 DB00091 L04AD01 Cyclosporine Cyclo~ A cyclic undecapeptide fro~ Enzyme--~
## 5 DB00091 L04AD01 Cyclosporine Cyclo~ A cyclic undecapeptide fro~ Antirhe~
```

The query results give some informations about drugs that are both codified according to DrugBank and the Anatomical Therapeutic and Chemical classification (ATC).

But querying knowledge and data through SPARQL endpoints on the Web requires an expertise in SPARQL syntax, the knowledge of potential useful SPARQL endpoints and also the representation of the knowledge they contain. This is probably why their use remains shy in some domains, for example in epidemiology and more generally in public health.

queryMed provides predefined SPARQL queries dedicated to medical and pharmacological domains –drugs and diseases– embeded in R functions.

Could we retrieve some information about the drugs present in a healthcare database ?

The example dataset *drug_set* is a dataframe that contains patients Id and prescribed drugs, codified according to the ATC.

```
data(drug_set)
drug_set[1:5,1:2]
```

```
##   patient     ATC
## 1       1 B01AC04
## 2       1 B01AC04
## 3       1 B01AC06
## 4       1 B01AC06
## 5       1 B03AA02
```

To retrieve some information about drugs we could call *bio2rdf()* or *dbpedia()*. These functions send predefined queries on Bio2RDF and DBpedia.

```
bio2rdf <- uri2norm(bio2rdf_db(lang="en"))
```

```
## Querying http://bio2rdf.org/sparql
```

```
dbpedia <- uri2norm(dbpedia_drug(lang="en"))
```

```
## Querying https://dbpedia.org/sparql
```

And then we could apply a filter on the drug present in our database :

```
drug_set_bio2rdf <- bio2rdf[bio2rdf$atc %in% drug_set$ATC,]
drug_set_dbpedia <- dbpedia[dbpedia$atc %in% drug_set$ATC,]
head(drug_set_bio2rdf)
```

```
## # A tibble: 6 x 6
##   db      atc     title        label  description                 category
##   <chr>   <chr>   <chr>        <chr>  <chr>                       <chr>
## 1 DB00091 L04AD01 Cyclosporine Cyclo~ A cyclic undecapeptide fro~ Antifun~
## 2 DB00091 L04AD01 Cyclosporine Cyclo~ A cyclic undecapeptide fro~ Enzyme-~
## 3 DB00091 L04AD01 Cyclosporine Cyclo~ A cyclic undecapeptide fro~ Antirhe~
## 4 DB00091 L04AD01 Cyclosporine Cyclo~ A cyclic undecapeptide fro~ Dermato~
## 5 DB00091 L04AD01 Cyclosporine Cyclo~ A cyclic undecapeptide fro~ Immunos~
## 6 DB00099 L03AA02 Filgrastim   Filgr~ Filgrastim is a recombinan~ Hematop~
```

```
head(drug_set_dbpedia)
```

```
## # A tibble: 6 x 6
##   drug          atc     db      abstract                 comment     label
##   <chr>         <chr>   <chr>   <chr>                    <chr>       <chr>
## 1 Clozapine     N05AH02 DB00363 Clozapine, sold under t~ Clozapine,~ Cloz~
## 2 Valproate     N03AG01 DB00313 Valproate (VPA), and it~ Valproate ~ Valp~
## 3 Sulfasalazine A07EC01 DB00795 Sulfasalazine (SSZ), ma~ Sulfasalaz~ Sulf~
## 4 Amitriptyline N06AA09 DB00321 Amitriptyline, sold und~ Amitriptyl~ Amit~
## 5 Ergotamine    N02CA02 DB00696 Ergotamine is an ergope~ Ergotamine~ Ergo~
## 6 Itraconazole  J02AC02 DB01167 Itraconazole (code name~ Itraconazo~ Itra~
```

But drugs are not always codified according to the ATC nomenclature. Linked Data initiatives have made significant efforts to provide links –or mappings– between the main nomenclatures. For example, the Concept Unique Identifier(CUI) from the Unified Medical Language System has been used to annotate codes of drugs and diagnoses from several nomenclatures. This kind of mapping is not always easy to use. We provide a function, *mapping_cui()*, that allows to search for a CUI mapping between medical terms. Because the function programmatically accesss to BioPortal API (Whetzel et al. 2011) to search for a potential mapping, it needs an API key. To have one, you need to register at BioPortal.

```
drug_ATC_NDFRT <- mapping_cui(codes=drug_set$ATC,
                              ontologies_source="ATC",
                              ontologies_target="NDFRT",
                              api_key="your_api_key")
head(drug_ATC_NDFRT)
```

It gives you a mapping between ATC codes from *drug_set* and the National Drug File - Reference Terminology (NDF-RT), using CUI, when it exists (an ATC or NDF-RT code usually leads to at least one CUI code, but a term in NDF-RT could not exist in ATC, or vice versa).

Then we can merge this mapping table to our initial database :

```
drug_set_ATC_CUI_NDFRT <- merge(drug_set[,c("patient","ATC")],
                                drug_ATC_NDFRT,by.x="ATC",
                                by.y="source",
                                all.x=T)
head(drug_set_ATC_CUI_NDFRT)
```

This allows to extend the approach to the main medical and pharmacological nomenclatures. For the next examples, we added NDF-RT mapping in *drug_set* and *disease_set* databases.

This is for the moment quite simple annotation. queryMed offers also the possibility to retrieve more complex informations, such as drug interactions, drug-disease contraindications and drug indications.

Drug-disease contraindications from the National Drug File - Reference Terminology

The *NDFRT_CI_with()* function send a SPARQL query on Ontobee SPARQL endpoint to retrieve contraindications between drugs and diseases :

```
NDFRT_CI <- NDFRT_CI_with()
```

```
## Querying http://sparql.hegroup.org/sparql/
```

```
NDFRT_CI <- uri2norm(NDFRT_CI)
head(NDFRT_CI)
```

```
## # A tibble: 6 x 6
##   ndf_drug    cui_drug label_drug       ndf_diag cui_diag label_diag
##   <chr>       <chr>    <chr>            <chr>    <chr>    <chr>
## 1 N0000020091 C0014704 ERGONOVINE       N000000~ C0000821 Abortion, Threa~
## 2 N0000145814 C0059514 ERGONOVINE MALE~ N000000~ C0000821 Abortion, Threa~
## 3 N0000023156 C1572765 WARFARIN SODIUM~ N000000~ C0000821 Abortion, Threa~
## 4 N0000022035 C0244656 FOSPHENYTOIN     N000000~ C0001396 Adams-Stokes Sy~
## 5 N0000022099 C0733758 FOLLITROPIN      N000000~ C0001621 Adrenal Gland D~
## 6 N0000145817 C0012258 DIGITOXIN        N000000~ C0002726 Amyloidosis [Di~
```

If SPARQL endpoints and medical ontologies are quite dispersed over the Web, some initiatives have tried to gather similar knowledge from different sources from the Linked Data. Hence, the Drug Indication Database (DID) have pooled twelve sources of knowledge about drug indications (Sharp 2017). Similarly, the Drug Interaction Knowledge Base (DIKB) have collected fourteen sources of knowledge about potential drug interactions (Ayvaz et al. 2015).

DID and DIKB

Curated versions of DID and DIKB are available in *queryMed* as build-in datasets.

```
data(DIKB)
data(DID)
```

We have now simple knowledge (e.g. definitions, synonyms, comments) as well as complex knowledge to annotate health data. If the simple knowledge is easy to merge with a health database of diseases or drugs,

complex knowledge such as contraindications, interactions or indications, needs a more complex function to search for semantic relations (here specifically pairs of codes) in a database.

find_relations() function aims to perform this kind of mining. And with the appropriate knowledge, it can help to answer the following questions :

- Do patients have drug-disease contraindications ?
- Do patients have drug interaction ?
- Do patients have drug indicated for their disease or health status ?

Let us answer to these questions on the test databases present in *queryMed* : *drug_set* and *disease_set*. Similarly to *drug_set*, *disease_set* is a test dataframe that contains diseases codes for patients, codified according to the International Classification of Diseases - 10th revision (ICD10), and mapped to CUI and NDF-RT.

```
data(disease_set)
head(disease_set)
```

```
##   patient ICD10      cui      NDF-RT
## 1       1 I73.9 C0021775 N0000001694
## 2       1 I73.9 C0085096 N0000003422
## 3       1 I73.9 C0085617        <NA>
## 4       2 I74.4 C0340579        <NA>
## 5       2 I74.4 C0564750        <NA>
## 6       3 I74.4 C0340579        <NA>
```

Do patients have drug-disease contraindications ?

NDF-RT with *find_relations()* can help answer this question :

```
contraindications <- find_relations(data.x=drug_set,
                                    data.y=disease_set,
                                    data_indices = "patient",
                                    data_elements.x = "NDF-RT",
                                    data_elements.y = "NDF-RT",
                                    target=NDFRT_CI,
                                    target_elements = c("ndf_drug","ndf_diag"),
                                    progress="none")

nb_contraindications <- sum(contraindications != "No known relations")
```

We identified 1 patient(s) having at least one drug-disease contraindication, according to NDF-RT.

```
contraindications[contraindications != "No known relations"][1]
```

```
## $`658`
## # A tibble: 1 x 6
##   ndf_drug    cui_drug label_drug ndf_diag    cui_diag label_diag
##   <chr>       <chr>    <chr>      <chr>       <chr>    <chr>
## 1 N0000020412 C0014710 ERGOTAMINE N0000003422 C0085096 Peripheral Vascula~
```

Do patients have drug interaction ?

DIKB can help answer this question :

```
interactions <- find_relations(data.x=drug_set,
                               data_indices = "patient",
                               data_elements.x = "ATC",
                               target=DIKB,
                               target_elements = c("atc1","atc2"),
                               progress="none")

nb_interact<- sum(interactions != "No known relations")
```

We identified 585 patients who have at least one drug interaction, according to DIKB. Here is an example :

```
interactions[interactions != "No known relations"][1]
```

```
## $`1`
##      drug2   drug1      object precipitant contraindication ddiPkMechanism
## 6  DB01118 DB05039 INDACATEROL  AMIODARONE            FALSE           <NA>
## 9  DB01118 DB00758 CLOPIDOGREL  AMIODARONE            FALSE           <NA>
## 10 DB05039 DB01118  AMIODARONE INDACATEROL            FALSE           <NA>
##    effectConcept label precaution severity  uri     source
## 6           <NA>  <NA>      FALSE     <NA> <NA>   Drugbank
## 9           <NA>  <NA>      FALSE     <NA> <NA> NLM-Corpus
## 10          <NA>  <NA>      FALSE     <NA> <NA>   Drugbank
##       evidenceStatement    atc1    atc2
## 6                  <NA> R03AC18 C01BD01
## 9  Specific_Interaction B01AC04 C01BD01
## 10                 <NA> C01BD01 R03AC18
```

Do patients have drug indicated for their disease or health status ?

DID can help answer this question :

```
indications <- find_relations(data.x=drug_set, data.y=disease_set,
                              data_indices = "patient",
                              data_elements.x= "ATC",
                              data_elements.y ="ICD10",
                              target=DID,
                              target_elements=c("atc","icd10"),
                              progress="none")

nb_indication <- sum(indications != "No known relations")
```

We identified 93 patients having at least one relation of indication between their drugs and their diseases, according to DID.

References

Ayvaz, Serkan, John Horn, Oktie Hassanzadeh, Qian Zhu, Johann Stan, Nicholas P. Tatonetti, Santiago Vilar, et al. 2015. "Toward a Complete Dataset of Drug–drug Interaction Information from Publicly Available Sources." *Journal of Biomedical Informatics* 55 (June): 206–17. doi:10.1016/j.jbi.2015.04.006.

Callahan, Alison, José Cruz-Toledo, Peter Ansell, and Michel Dumontier. 2013. "Bio2RDF Release 2: Improved Coverage, Interoperability and Provenance of Life Science Linked Data." In *The Semantic Web:*

Semantics and Big Data, 200–212. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-38288-8_14.

Ferreira, João D, Daniela Paolotti, Francisco M Couto, and Mário J Silva. 2013. "On the Usefulness of Ontologies in Epidemiology Research and Practice." *Journal of Epidemiology and Community Health* 67 (5): 385–88. doi:10.1136/jech-2012-201142.

Lehmann, Jens, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, et al. 2015. "DBpedia - A Large-Scale, Multilingual Knowledge Base Extracted from Wikipedia." *Undefined*. /paper/DBpedia-A-large-scale%2C-multilingual-knowledge-base-Lehmann-Isele/4fa0d9c4c3d17458085ee255b7a4b7c325d59e32.

Ong, Edison, Zuoshuang Xiang, Bin Zhao, Yue Liu, Yu Lin, Jie Zheng, Chris Mungall, Mélanie Courtot, Alan Ruttenberg, and Yongqun He. 2017. "Ontobee: A Linked Ontology Data Server to Support Ontology Term Dereferencing, Linkage, Query and Integration." *Nucleic Acids Research* 45 (D1): D347–D352. doi:10.1093/nar/gkw918.

Pathak, Jyotishman, Richard C. Kiefer, and Christopher G. Chute. 2013. "Using Linked Data for Mining Drug-Drug Interactions in Electronic Health Records." *Studies in Health Technology and Informatics* 192: 682–86.

Salvadores, Manuel, Paul R. Alexander, Mark A. Musen, and Natalya F. Noy. 2013. "BioPortal as a Dataset of Linked Biomedical Ontologies and Terminologies in RDF." *Semantic Web* 4 (3): 277–84.

Sharp, Mark E. 2017. "Toward a Comprehensive Drug Ontology: Extraction of Drug-Indication Relations from Diverse Information Sources." *Journal of Biomedical Semantics* 8 (1). doi:10.1186/s13326-016-0110-0.

Whetzel, Patricia L., Natalya F. Noy, Nigam H. Shah, Paul R. Alexander, Csongor Nyulas, Tania Tudorache, and Mark A. Musen. 2011. "BioPortal: Enhanced Functionality via New Web Services from the National Center for Biomedical Ontology to Access and Use Ontologies in Software Applications." *Nucleic Acids Research* 39 (Web Server issue): W541–545. doi:10.1093/nar/gkr469.