Using CemConnector

2021-07-27

Contents

If you're reading this guide it is expected that you have some familiarity with the common evidence model (CEM) and have an installation set up and configured. In the following example we will observe all of the relevant evidence for the drug ingredient `codine` with the outcome of `depression`.

Using CEMConnector with a web backend

The simplest way to connect with the CEM is to connect to a host web instance, for example the public API available at https://cem.ohdsi.org/ (TBD). To do this we will use a web backend. First we need to create a connection to the web api:

```r
cemConnection <- CemConnector::CemWebApiBackend(apiUrl = "https://cem.ohdsi.org/")
```

Using CEMConnector with a database backend

Alternatively, a connection can be made with a DatabaseConnector::connectionDetails object. Contact your CEM administrator for details connecting this way.

```r
connectionDetails <- DatabaseConnector::createConnectionDetails(user = "mydbusername",
                                                                server = "myserver/foo",
                                                                dbms = "redshift",
                                                                password = "mysecret")

cemConnection <- CemConnector::CemDatabaseBackend$new(connectionDetails = connectionDetails,
                                                      vocabularySchema = "vocabulary",
                                                      cemSchema = "cem_v2",
                                                      sourceSchema = "cem_v2_source")
```

This should connect to the server without error. If you get an error please check that the URL is valid, you may need to contact your administrator.

Constructing conceptset defintions

To generate concept sets we recommend using PHEOBE. This is a simple way of finding standard concepts (SNOMED or RxNorm) that relate to diseases and outcomes of interest for which real world evidence has been observed. For this example we will use the concept for moderate depression as a disease outcome and the ingredient exposure codeine. It is important to note that the CEM stores information in the standard vocabulary only. Similarly drug concepts are mapped at the ingredient level, so you will want to map specific ingredient concepts not drug formulations. For example, instead of using a specific drug dose such as 'codeine 30 MG' it would be advisable to use only the ingredient 'codeine'. Naturally, designing concept sets is highly specific to a phenotype of interest and careful consideration should go into the concepts used.

In this example, we would like to both find any existing evidence that relates these two diseases as well as select potential negative controls that can be used to calibrate p-values and confidence intervals from effect estimates that we may generate in observational studies.

Searching for outcomes

The simplest drug concept set is of one ingredient.

```
ingreditentConceptSet <- data.frame(conceptId = c(1201620), includeDescendants = c(0), isExcluded = c(0
```

Though not necessary for this ingredient set, we allow the utility to look for descendants. This may be useful for an entire class of medications such as the ATC classification 'Other opioids':

```
parentIngredientSet <- data.frame(conceptId = c(21604296), includeDescendants = c(1), isExcluded = c(0)
```

We would like to search for any related disease concepts to this query. Cem connector provides a simple evidence summary for all resulting concepts that are in the database.

```
outcomeConcepts <- cemConnection$getIngredientEvidenceSummary(ingreditentConceptSet)
outcomeConcepts
#> # A tibble: 0 x 3
#> # ... with 3 variables: conditionConceptId <int>, conceptName <chr>,
#> #   evidenceExists <lgl>
```

The resulting data frame is of ingredient concepts with a 1 or 0 if any evidence exists within the CEM.

For an item here, we may want to look at the specific evidence that is involved. To do this we would look at an exposure outcome pair from the concept set.

To get all the evidence found in the CEM for the ingredient concept set, what source its from and what the counts or statistics are, we can look at the entire search universe.

```
outcomeConcepts <- cemConnection$getIngredientEvidenceSummary(ingreditentConceptSet)
```

Negative control suggestions

Cem Connector is capable of suggesting negative controls for use in population level effect estimate studies. Negative controls are intended to be exposures or outcomes selected because they are believed to have no direct causal relationship with the outcome of interest. For example, in-grown toe-nails are not caused by exposure to ACE inhibitors. However, in-grown toenails are extremely common, and many patients will experience this outcome while exposed to the drug. Consequently, any observed effect is a product of residual bias (e.g. from the study design choices) and this can be used to produce calibrated effect estimates.

The design of a PLE study is outside the scope of this viggnette, however, the selection of outcomes/exposures can be achieved at the concept level. Here, the CemConnector takes a concept set as a parameter and finds outcomes our exposures that have no mapped relationships within the CEM. We note that these mappings are not perfect and human curation by experienced clinicians is advised.

To find negative control outcomes for the above ingredient set use the following code.

```
outcomeConcepts <- cemConnection$getSuggestedControlCondtions(ingreditentConceptSet)
outcomeConcepts
#> # A tibble: 0 x 2
#> # ... with 2 variables: conditionConceptId <int>, conceptName <chr>
```

Searching for evidence pairs

It is often desirable to find out what information relates exposures and outcomes.