Working with clinical code lists with rEHR

David A Springate[*1], *Ivan Olier*[2], *Evangelos Kontopantelis*[1]
[1]*Institute of Population Health, University of Manchester*
[2]*Institute of Biotechnology, University of Manchester*
[*]*Corresponding Author*

2016-01-05

Abstract

The rEHR package provides functions to identify relevant code lists and to automate the construction of clinical code lists.

The Package is available on CRAN.

```
install.packages("rEHR")
```

Alternatively, the development verion of the `rEHR` package can be downloaded from Github.

```
install.packages("devtools")
require(devtools)
install_github("rEHR", "rOpenHealth")
require(rEHR)
```

Building draft definition lists

Definition lists can be defined for:

- clinical terms (by either text search or searching for matching clinical codes)
- test terms (by text search)
- medications (by either text search or product code)

Building definition lists is a two stage process:

1. The search is defined by instantiating an object of class `MedicalDefinition`, containing the terms to be searched for in the lookup tables
2. A `definition_search` is performed on the `MedicalDefinition` object and the relevant lookup tables to return a list of matching dataframes

A `MedicalDefinition` object can be either made using terms defined within `R` or with terms imported from an external csv file

Defining searches within R

Use the `MedicalDefinition` constructor function to generate search definitions. This takes the following arguments:

- `terms` a list of character vectors representing clinical search terms or NULL

- `codes` list of character vectors representing clinical code terms or NULL
- `tests` list of character vectors representing test search terms or NULL
- `drugs` list of character vectors representing drug search terms or NULL
- `drugcodes` list of character vectors representing drug product code terms or NULL

vectors of length > 1 are searched for together (AND), in any order. Different vectors in the same list are searched for seperately (OR). Placing a "-" character at the start of a character vector element excludes that terms from the search.

```
# vectors of length > 1 are combined as a single AND expression
# "-" excludes that term from the search
def <- MedicalDefinition(terms = list("peripheral vascular disease", "peripheral gangrene", "-wrong ans
                         "intermittent claudication", "thromboangiitis obliterans",
                         "thromboangiitis obliterans", "diabetic peripheral angiopathy",
                         c("diabetes", "peripheral angiopathy"),  # combined as a single AND expression
                         c("diabetes", "peripheral angiopathy"),
                         c("buerger",  "disease presenile_gangrene"),
                         "thromboangiitis obliterans",
                         "-rubbish", # exclusion
                         c("percutaneous_transluminal_angioplasty", "artery"),
                         c("bypass", "iliac_artery"),
                         c("bypass", "femoral_artery"),
                         c("femoral_artery" , "occlusion"),
                         c("popliteal_artery", "occlusion"),
                         "dissecting_aortic_aneurysm", "peripheral_angiopathic_disease",
                         "acrocyanosis", "acroparaesthesia", "erythrocyanosis",
                         "erythromelalgia", "ABPI",
                         c("ankle", "brachial"),
                         c("ankle", "pressure"),
                         c("left", "brachial"),
                         c("left", "pressure"),
                         c("right", "brachial"),
                         c("right", "pressure")),
         codes = list("G73"),
         tests = NULL,
         drugs = list("insulin", "diabet", "aspirin"))
```

When searching for codes, a range of clinical codes can be searched for by providing two codes seperated by a hyphen. e.g. `E114-E117z`.

importing searches via a csv file

Searches can be imported from a csv file in this format

The first column in every row determines the list that the term applies to and the second column determines whether the term should be included or excluded. Note that the csv does not have to be a valid format for conversion to a dataframe. Extra columns can be used to include terms to be combined as an AND expression with the other terms on that row. The title row can also be ommitted. You can use standard regex escape patterns in the term definitions.

The data is called into `R` in the following way:

```
## Using the example search definition provided with the package
def2 <- import_definitions(system.file("extdata", "example_search.csv", package = "rpcdsearch"))
```

Running searches

Once a search has been defined, the relevant lookup tables should be called in. Note that these lookup tables are not provided with the package and will be specific to the users EHR database. These examples are using CPRD lookups and EHR definitions (See the ehr_system code for details of how the interface with CPRD is implemented).

```
## Use fileEncoding="latin1" to avoid any issues with non-ascii characters
medical_table <- read.delim("Lookups//medical.txt", fileEncoding="latin1", stringsAsFactors = FALSE)
drug_table <- read.delim("Lookups/product.txt", fileEncoding="latin1", stringsAsFactors = FALSE)
```

And the search can be run:

```
draft_lists <- build_definition_lists(def, medical_table = medical_table,drug_table = drug_table)
```

This returns a list of dataframes for each of the provided search lists. If `terms` and `codes` are provided in the definition, it also contains a `combined_terms_codes` data frame which is a combination of `terms` and `codes` with duplicate rows removed.

Exporting code lists

The code lists produced by `build_definition_lists` will often want to be reviewed by clinicians or non-technical researchers. To facilitate this, there is an `export_definition_search` function to export the code lists as an Excel file, with each list occupying a tab in the file. To export a code list:

```
out_file <- "def_searches.xlsx"
export_definition_search(draft_lists, out_file)
```