CPRDTools: a CPRD GOLD data wrangling toolkit

Abstract

The analysis of large scale data, and moreover the analysis of large scale electronic health records data is become more commonplace. The ease and ability of modern data generation and capture means there is the potential across almost all industries to capture more data now, than ever before and that equates to large data resources, with CPRD no exception. These large data resources, as attractive and appealing as they may be to researchers, pose a significant problem - how to manage all that data? CPRDTools is a collection of wrapper R functions intended to simplify the loading, extraction and management of Clinical Practice Research Datalink (CPRD) GOLD specific and associated electgronic health records data. Allowing for the loading of CPRD and non-CPRD data into a SQLite based database, providing an efficient, secure and updatable repository for the data, keeing the origional source files intact. Through data queries, user-defined data are drawn allowing for subsetting, joining and filtering in a signle step, creating alaysis ready data.

Introduction

The CPRD is one of the largest longitudinal medical records databases in the world, supported by the National Institute of Health (NIHR) and the Medicines and Healthcare products Regulatory Agency (MHRA). It was first established in 1987 as the Value Added Medical Products (VAMP) dataset, this grew into the General Practice Research Database (GPRD) in 1993, before its final transition into CPRD in 2012 (Herrett et al. 2015).

PRD GOLD data are comprised of ten separate datasets: patient, practice, staff, consultation, clinical, additional clinical details, referral, immunisation, test and therapy (Padmanabhan 2017). These datasets contain their specific data and are linkable through a unique linkage key field, where key does not imply importance but a unique variable contained in two datasets allowing them to be joined, such as the CPRD-assigned and anonymised unique patient identifier `patid`.

Due to the size of CPRD, data extracted for research are spread over multiple text (`.txt`) files within each dataset to enable file transfer. This means that for CPRD clinical data, for instance, a researcher may receive their requested clinical data broken up over 25 individual text files. This is done to aid with file completeness and reduce turn-around times if errors are found. If an error occurred during the transfer of data between the data owner and the researcher or in the extraction of the requested data by the data owner, the error can potentially be limited to only select files, requiring only their replacement with the corrected/error-free versions.

Often these text files are additionally zipped, or compressed, requiring that these files first be uncompressed or unzipped. These multiple files from each dataset (clinical, referral etc.) then need to be grouped together and amalgamated into a single `table`. Tables are a collection of data of the same shape, from the same dataset. In CPRD, each separate dataset (patient, practice, clinical etc.) forms a table. These tables are then stored in the SQLite database.

SQLite is an opensource, SQL based database engine (SQLite 2017). The use of a SQLite database provides an efficient storage solution, allowing for the loading, updating and maintenance of the database, all while retaining the original *raw* data files unaltered. An SQlite database permits for rapid data extraction through the use of data queries, drawing the required data from the database, allowing filtering, sub-setting, limiting and the joining of data in a single execution step.

This document aims to provide a simple and introductory overview of `CPRDTools` and its application to

arbitrary (fictional) data. This data though are provided in the manner in which many CPRD data extract are received, where data are spread over multiple files and often located in sub-folders.

CPRDTools are loaded using:

```
library(devtools)
install_github("JamesCFSchmidt/CPRDTools")
```

```
##
## * checking for file '/local/jcfs2/Rtmp2CnlK2/remotes4d1a5eb9e740/JamesCFSchmidt-CPRDTools-dd6791a/DES
## * preparing 'CPRDTools':
## * checking DESCRIPTION meta-information ... OK
## * checking for LF line-endings in source and make files and shell scripts
## * checking for empty or unneeded directories
## Omitted 'LazyData' from DESCRIPTION
## * building 'CPRDTools_0.0.0.9000.tar.gz'
```

```
library(CPRDTools)
```

CPRDTools overview

The functions within 'CPRDTools' broadly fall into three groups, categorised by their general application area: (1) - loading, (2) - maintenance and other tasks and (3) - extraction. **Loading** encompasses all functions used in the reading, converting and writing of data into the database including a function used to list all available files and all available CPRD files in a specified location, functions used in database maintenance, query speed improvements and date conversion functions fall under **maintenance and other tasks**, and finally, functions used to, and in the process of, drawing and retrieving data from the database fall within **extraction**.

The data

Loading

Before loading any data into the database, it is best practice to understand the layout of the *raw* data. This can be achieved by either navigating to the location where the data are stored or by employing the `list_files` and `list_cprd` functions.

Listing available data

The `list_files` function provides a list of all files of a specified type in a specified location. To view all text (`.txt`) files in the data location

```
list_files(file_location = FILE.path,
           file_type = ".txt")
```

```
## $file_location
## [1] "/rfs/LRWE_Proj59/jcfs2/Test"
##
## $files
##                                                   files
## 1 /rfs/LRWE_Proj59/jcfs2/Test/hes_patient_19_253R.txt
```

To view all compressed/*zipped* (`.zip`) files in the data location

```
list_files(file_location = FILE.path,
           file_type = ".zip")
```

```
## $file_location
## [1] "/rfs/LRWE_Proj59/jcfs2/Test"
##
## $files
##                                                        files
## 1 /rfs/LRWE_Proj59/jcfs2/Test/305546.ms2_Extract_Clinical_001.zip
## 2 /rfs/LRWE_Proj59/jcfs2/Test/305546.ms2_Extract_Clinical_002.zip
## 3 /rfs/LRWE_Proj59/jcfs2/Test/305546.ms2_Extract_Clinical_003.zip
## 4 /rfs/LRWE_Proj59/jcfs2/Test/305546.ms2_Extract_Clinical_004.zip
## 5 /rfs/LRWE_Proj59/jcfs2/Test/305720.ms1_Extract_Practice_001.zip
## 6 /rfs/LRWE_Proj59/jcfs2/Test/305720.ms2_Extract_Practice_001.zip
```

And to view all files in the data location, excluding files in sub-folders

```
list_files(file_location = FILE.path,
           file_type = "all")
```

```
## $file_location
## [1] "/rfs/LRWE_Proj59/jcfs2/Test"
##
## $files
##                                                          files
## 1  /rfs/LRWE_Proj59/jcfs2/Test/305546.ms2_Extract_Clinical_001.zip
## 2  /rfs/LRWE_Proj59/jcfs2/Test/305546.ms2_Extract_Clinical_002.zip
## 3  /rfs/LRWE_Proj59/jcfs2/Test/305546.ms2_Extract_Clinical_003.zip
## 4  /rfs/LRWE_Proj59/jcfs2/Test/305546.ms2_Extract_Clinical_004.zip
## 5  /rfs/LRWE_Proj59/jcfs2/Test/305720.ms1_Extract_Practice_001.zip
## 6  /rfs/LRWE_Proj59/jcfs2/Test/305720.ms2_Extract_Practice_001.zip
## 7                         /rfs/LRWE_Proj59/jcfs2/Test/CCI.xlsx
## 8             /rfs/LRWE_Proj59/jcfs2/Test/hes_patient_19_253R.txt
## 9                      /rfs/LRWE_Proj59/jcfs2/Test/ons_lt.csv
## 10                       /rfs/LRWE_Proj59/jcfs2/Test/patient
```

From the above it can be seen that there are six zipped text files (`.zip`), one excel file (`.xlsx`), one standard text file (`.txt`) and a folder, `patient`. In order to view CPRD GOLD specific files, corresponding to names CPRD GOLD datasets (*patient, practice, staff, consultation, clinical, additional clinical details, referral, immunisation, test and therapy*), the `list_cprd` function is used. This function provides the core functionality to the loading of CPRD GOLD data, generating a list of CPRD GOLD specific files in the specified location.

```
list_cprd(file_location = FILE.path,
          folder = F,
          zip = T)
```

```
## $file_location
## [1] "/rfs/LRWE_Proj59/jcfs2/Test"
##
## $all_files_tables
##                                                          files    table
## 1 /rfs/LRWE_Proj59/jcfs2/Test/305546.ms2_Extract_Clinical_001.zip Clinical
## 2 /rfs/LRWE_Proj59/jcfs2/Test/305546.ms2_Extract_Clinical_002.zip Clinical
## 3 /rfs/LRWE_Proj59/jcfs2/Test/305546.ms2_Extract_Clinical_003.zip Clinical
## 4 /rfs/LRWE_Proj59/jcfs2/Test/305546.ms2_Extract_Clinical_004.zip Clinical
## 5 /rfs/LRWE_Proj59/jcfs2/Test/305720.ms1_Extract_Practice_001.zip Practice
```

```
## 6 /rfs/LRWE_Proj59/jcfs2/Test/305720.ms2_Extract_Practice_001.zip Practice
##
## $tables
##      Table File_Count
## 1 Clinical         4
## 2 Practice         2
```

Here the folders input is defined as FALSE, showing only zipped data contained in the data location specified. When the folder input is TRUE, the available data found in the `patient` folder is displayed.

```
list_cprd(file_location = FILE.path,
          folder = T,
          zip = T)
```

```
## $file_location
## [1] "/rfs/LRWE_Proj59/jcfs2/Test"
##
## $all_files_tables
##                                                                files
## 1 /rfs/LRWE_Proj59/jcfs2/Test/patient/305508.ms1_Extract_Patient_001.zip
## 2 /rfs/LRWE_Proj59/jcfs2/Test/patient/305508.ms2_Extract_Patient_001.zip
##      table
## 1 Patient
## 2 Patient
##
## $tables
##      Table File_Count
## 1 Patient         2
```

In the `$table` output, the `list_cprd` function defines the tables and the count of *raw* files associated with that table in the data location. This outputted list is core to the automated loading of CPRD data, using the tables and files in each table to load all or a selection of specified CPRD GOLD files.

Loading all available data

The `load_cprd` function, relying on the `list_cprd` function for file and table list from a specified data location is able to automatically load in a selection or all available CPRD GOLD files in a specified location. Importing, uncompressing (unzipping) and appending all files from a CPRD GOLD-specific table into a database.

To load the `patient` data into a new database, the `load_cprd` function is defined as

```
load_cprd(db_path = DB.path,
          file_location = FILE.path,
          tables_to_load = "Patient",
          folder = T,
          zip = T,
          load_mapping = F,
          overwrite = F)
```

```
## Rows: 12 Columns: 20
## -- Column specification --------------------------------------------------
## Delimiter: "\t"
## chr  (4): frd, crd, tod, deathdate
## dbl (14): patid, vmid, gender, yob, marital, famnum, CHSreg, prescr, capsup,...
## lgl  (2): mob, CHSdate
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## ----------LOAD OF TABLE Patient, FILE No. 1 SUCCESSFUL----------

##
## Rows: 12 Columns: 20-- Column specification ----------------------------------------------------------
## Delimiter: "\t"
## chr  (4): frd, crd, tod, deathdate
## dbl (14): patid, vmid, gender, yob, marital, famnum, CHSreg, prescr, capsup,...
## lgl  (2): mob, CHSdate
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## ----------LOAD OF TABLE Patient, FILE No. 2 SUCCESSFUL----------

##

## $database_location
## [1] "/rfs/LRWE_Proj59/jcfs2/Test"
##
## $files_in_location
##                                                                       files
## 1 /rfs/LRWE_Proj59/jcfs2/Test/patient/305508.ms1_Extract_Patient_001.zip
## 2 /rfs/LRWE_Proj59/jcfs2/Test/patient/305508.ms2_Extract_Patient_001.zip
##     table
## 1 Patient
## 2 Patient
##
## $tables_in_location
##     Table File_Count
## 1 Patient          2
##
## $loaded_tables
## [1] "Patient"
##
## $load_report
##     table load_total file_total size_Mb size_Gb
## 1 Patient          2          2   0.001       0
##
## $database_tables
## [1] "Patient"
##
## $load_start
## [1] "2022-07-11 13:35:59 BST"
##
## $load_end
## [1] "2022-07-11 13:36:00 BST"
##
## $load_time
## Time difference of 0.9927347 secs
```

The `patient` data are found in a sub-folder within the data location () and the data are compressed (). As this is a new load of the database, no overwriting is required () and no CPRD specific mapping is needed ().

Alternatively to specifying each individual table to load (*additional, clinical, consultation, immunisation,*

patient, practice, referral, staff, test, therapy), the specification of the loading of *all* tables can be performed.

```
load_cprd(db_path = DB.path,
          file_location = FILE.path,
          tables_to_load = "all",
          folder = F,
          zip = T,
          load_mapping = F,
          overwrite = F)
```

This returns an error

```
## Error in load_cprd(db_path = DB.path, file_location = FILE.path, tables_to_load = "all", : tables_to
```

The error is due to the `patient` table already being contained in the database. The should only be applied on new databases or in conjunction with . The remainder of the CPRD GOLD data are loaded with

```
load_cprd(db_path = DB.path,
          file_location = FILE.path,
          tables_to_load = c("Clinical","Practice"),
          folder = F,
          zip = T,
          load_mapping = F,
          overwrite = F)
```

```
## Rows: 12 Columns: 10
## -- Column specification ------------------------------------------------------
## Delimiter: "\t"
## chr (2): eventdate, sysdate
## dbl (8): patid, constype, consid, medcode, staffid, episode, enttype, adid
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.

## ----------LOAD OF TABLE Clinical, FILE No. 1 SUCCESSFUL----------

##
## Rows: 12 Columns: 10-- Column specification ---------------------------------------------------------
## Delimiter: "\t"
## chr (2): eventdate, sysdate
## dbl (8): patid, constype, consid, medcode, staffid, episode, enttype, adid
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.

## ----------LOAD OF TABLE Clinical, FILE No. 2 SUCCESSFUL----------

##
## Rows: 12 Columns: 10-- Column specification ---------------------------------------------------------
## Delimiter: "\t"
## chr (2): eventdate, sysdate
## dbl (8): patid, constype, consid, medcode, staffid, episode, enttype, adid
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.

## ----------LOAD OF TABLE Clinical, FILE No. 3 SUCCESSFUL----------

##
## Rows: 12 Columns: 10-- Column specification ---------------------------------------------------------
## Delimiter: "\t"
## chr (2): eventdate, sysdate
```

```
## dbl (8): patid, constype, consid, medcode, staffid, episode, enttype, adid
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.

## ----------LOAD OF TABLE Clinical, FILE No. 4 SUCCESSFUL----------

##
## Rows: 5 Columns: 4-- Column specification ----------------------------------------------------------
## Delimiter: "\t"
## chr (2): lcd, uts
## dbl (2): pracid, region
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.

## ----------LOAD OF TABLE Practice, FILE No. 1 SUCCESSFUL----------

##
## Rows: 5 Columns: 4-- Column specification ----------------------------------------------------------
## Delimiter: "\t"
## chr (2): lcd, uts
## dbl (2): pracid, region
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.

## ----------LOAD OF TABLE Practice, FILE No. 2 SUCCESSFUL----------

##
## $database_location
## [1] "/rfs/LRWE_Proj59/jcfs2/Test"
##
## $files_in_location
##                                                       files    table
## 1 /rfs/LRWE_Proj59/jcfs2/Test/305546.ms2_Extract_Clinical_001.zip Clinical
## 2 /rfs/LRWE_Proj59/jcfs2/Test/305546.ms2_Extract_Clinical_002.zip Clinical
## 3 /rfs/LRWE_Proj59/jcfs2/Test/305546.ms2_Extract_Clinical_003.zip Clinical
## 4 /rfs/LRWE_Proj59/jcfs2/Test/305546.ms2_Extract_Clinical_004.zip Clinical
## 5 /rfs/LRWE_Proj59/jcfs2/Test/305720.ms1_Extract_Practice_001.zip Practice
## 6 /rfs/LRWE_Proj59/jcfs2/Test/305720.ms2_Extract_Practice_001.zip Practice
##
## $tables_in_location
##      Table File_Count
## 1 Clinical          4
## 2 Practice          2
##
## $loaded_tables
## [1] "Clinical" "Practice"
##
## $load_report
##      table load_total file_total size_Mb size_Gb
## 1 Clinical          4          4   2e-03       0
## 2 Practice          2          2   5e-04       0
##
## $database_tables
## [1] "Clinical" "Patient"  "Practice"
##
## $load_start
```

```
## [1] "2022-07-11 13:36:00 BST"
##
## $load_end
## [1] "2022-07-11 13:36:03 BST"
##
## $load_time
## Time difference of 2.962085 secs
```

The `$loaded_tables` output shows the current tables loaded and available in the database '

Loading a single CPRD table

The `patient` table previously could have been loaded singularly using the `load_table` function. Similar to the `load_cprd` function, this will load CPRD-specific data tables, using the compiled list of available files and their respective tables generated in the `list_cprd` function. This function imports, unzips and appends the files specific to a CPRD GOLD table but performs the task on a single table at a time.

This may be of use when the data are stored in separate file locations, not within sub-folders in the same data location. The loading of the `practice` information is performed using

```
load_table(db_path = DB.path,
           file_location = FILE.path,
           table_name = "Practice",
           zip = T,
           overwrite = T)
```

```
## -------Overwrite = TRUE, DELETION OF Practice COMPLETE-------
##
## Rows: 5 Columns: 4
## -- Column specification ------------------------------------------------
## Delimiter: "\t"
## chr (2): lcd, uts
## dbl (2): pracid, region
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.

## ----------LOAD OF TABLE Practice, FILE No. 1 SUCCESSFUL----------
##
## Rows: 5 Columns: 4-- Column specification -------------------------------------------------------
## Delimiter: "\t"
## chr (2): lcd, uts
## dbl (2): pracid, region
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.

## ----------LOAD OF TABLE Practice, FILE No. 2 SUCCESSFUL----------
##
## $database_location
## [1] "/rfs/LRWE_Proj59/jcfs2/Test"
##
## $files_in_location
##                                                             files    table
## 1 /rfs/LRWE_Proj59/jcfs2/Test/305546.ms2_Extract_Clinical_001.zip Clinical
```

```
## 2 /rfs/LRWE_Proj59/jcfs2/Test/305546.ms2_Extract_Clinical_002.zip Clinical
## 3 /rfs/LRWE_Proj59/jcfs2/Test/305546.ms2_Extract_Clinical_003.zip Clinical
## 4 /rfs/LRWE_Proj59/jcfs2/Test/305546.ms2_Extract_Clinical_004.zip Clinical
## 5 /rfs/LRWE_Proj59/jcfs2/Test/305720.ms1_Extract_Practice_001.zip Practice
## 6 /rfs/LRWE_Proj59/jcfs2/Test/305720.ms2_Extract_Practice_001.zip Practice
##
## $tables_in_location
##      Table File_Count
## 1 Clinical          4
## 2 Practice          2
##
## $loaded_table
## [1] "Practice"
##
## $load_report
##      table load_total file_total size_Mb size_Gb
## 1 Clinical          2          2   5e-04       0
## 2 Practice          2          2   5e-04       0
##
## $database_tables
## [1] "Clinical" "Patient"  "Practice"
##
## $load_start
## [1] "2022-07-11 13:36:03 BST"
##
## $load_end
## [1] "2022-07-11 13:36:03 BST"
##
## $load_time
## Time difference of 0.3906891 secs
```

Here as the practice table was previously loaded.

Updating a CPRD GOLD table

If you were to receive a new month or year of `patient` data, this would be appened to the data already contained in the database using

```
update_table(db_path = DB.path,
             file_location = "/rfs/LRWE_Proj59/jcfs2/Test/patient/305508.ms1_Extract_Patient_001.zip",
             table_name = "PaTient")
```

Note that capitalisation of the table name for CPRD-specific tables in unimportant. In general, SQL and therefore SQLite is not case sensitive.

Updating an additional file

For peripheral or additional data, whether CPRD or non-CPRD specific, the use of the `load_additional` function is employed. This function loads a singular file into a table (new or existing).

```
load_additional(db_path = DB.path,
                file_location = "/rfs/LRWE_Proj59/jcfs2/Test/hes_patient_19_253R.txt",
                type = ".txt",
```

```
                        table_name = "hes_patient",
                        overwrite = F)
```

```
## Rows: 24 Columns: 5
## -- Column specification ------------------------------------------------------
## Delimiter: "\t"
## dbl (5): patid, pracid, gen_hesid, n_patid_hes, match_rank
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## ----------LOAD OF TABLE hes_patient SUCCESSFUL----------

##

## $database_location
## [1] "/rfs/LRWE_Proj59/jcfs2/Test"
##
## $loaded_table
## [1] "hes_patient"
##
## $load_report
##         table size_Mb size_Gb
## 1 hes_patient   6e-04       0
##
## $database_tables
## [1] "Clinical"    "Patient"     "Practice"    "hes_patient"
##
## $load_start
## [1] "2022-07-11 13:36:03 BST"
##
## $load_end
## [1] "2022-07-11 13:36:03 BST"
##
## $load_time
## Time difference of 0.1077106 secs
```

The accepted files types are `.txt, .csv, .dta, .rds, .excel, .xl, .xls` and `.xlsx` with no additional input for file importing.

Updating an additional file

If sub-setting, manipulation or sheet selection, for instance, is required prior to the loading of a file into a table (new or existing) is required, it is first recommended to import the file in the usual manner into R. Once the data are in the format and layout that is required, the `load_global` function can be employed

```
## first import the life table data from .csv
life_table <- read.csv("/rfs/LRWE_Proj59/jcfs2/Database/Final.Datasets/ons_lt.csv")
##sub-set the data to the required fields
life_table <- life_table[,c("gender","year","age","pop_rate")]
## load into database
load_global(db_path = DB.path,
            file_to_load = life_table,
            table_name = "Life_table",
            overwrite = F)
```

```
## ----------LOAD OF TABLE Life_table SUCCESSFUL----------
##
## $database_location
## [1] "/rfs/LRWE_Proj59/jcfs2/Test"
##
## $loaded_table
## [1] "Life_table"
##
## $load_report
##        table size_Mb size_Gb
## 1 Life_table  0.1722   2e-04
##
## $database_tables
## [1] "Clinical"    "Life_table"  "Patient"     "Practice"     "hes_patient"
##
## $load_start
## [1] "2022-07-11 13:36:03 BST"
##
## $load_end
## [1] "2022-07-11 13:36:03 BST"
##
## $load_time
## Time difference of 0.07685947 secs
```

Extracting

With the database successfully loaded with our CPRD GOLD and associated data, we are now able to perform data extracts from the database. Before executing data extracts, it is *best practice* to first interrogate the databases for loaded table (ignoring that this is provided after each load iteration previously) and the for the structure and layout of a selection of tables. ### List tables in the database In order to list all loaded tables in the database, similar to the output in the `$loaded_tables` ## Maintenace and other tasks

References

Herrett, Emily, Arlene M. Gallagher, Krishnan Bhaskaran, Harriet Forbes, Rohini Mathur, Tjeerd van Staa, and Liam Smeeth. 2015. "Data Resource Profile: Clinical Practice Research Datalink (CPRD)." *International Journal of Epidemiology* 44 (3): 827–36. https://doi.org/10.1093/ije/dyv098.

Padmanabhan, Shivani. 2017. "CPRD Gold Data Specification." https://cprdcw.cprd.com/_docs/CPRD_GOLD_Full_Data_Specification_v2.0.pdf.

SQLite. 2017. *SQLite*. Charlotte, North Caolina: Hipp, Wyrick and Company, Inc. https://www.sqlite.org/index.html.