# Contents

**1 Getting Started** **1**

1.1 Developers: How to Add or Modify Analyses . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

1.2 License . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

# 1 Getting Started

(Please review the Achilles Wiki for specific details for Linux)

1. Make sure you have your data in the OMOP CDM v5.x format (https://github.com/OHDSI/CommonDataModel).

2. This package makes use of rJava. Make sure that you have Java installed. If you don't have Java already installed on your computer (on most computers it already is installed), go to java.com to get the latest version. If you are having trouble with rJava, this Stack Overflow post may assist you when you begin troubleshooting.

3. In R, use the following commands to install Achilles.

```
if (!require("devtools")) install.packages("devtools")

# To install the master branch
devtools::install_github("OHDSI/Achilles")

# To install latest release (if master branch contains a bug for you)
# devtools::install_github("OHDSI/Achilles@*release")

# To avoid Java 32 vs 64 issues
# devtools::install_github("OHDSI/Achilles", args="--no-multiarch")
```

4. To run the Achilles analysis, first determine if you'd like to run the function in multi-threaded mode or in single-threaded mode. Use `runCostAnalysis = FALSE` to save on execution time, as cost analyses tend to run long.

   **In multi-threaded mode**

   The analyses are run in multiple SQL sessions, which can be set using the `numThreads` setting and setting scratchDatabaseSchema to something other than `#`. For example, 10 threads means 10 independent SQL sessions. Intermediate results are written to scratch tables before finally being combined into the final results tables. Scratch tables are permanent tables; you can either choose to have Achilles drop these tables (`dropScratchTables = TRUE`) or you can drop them at a later time (`dropScratchTables = FALSE`). Dropping the scratch tables can add time to the full execution. If desired, you can set your own custom prefix for all Achilles analysis scratch tables (tempAchillesPrefix) and/or for all Achilles Heel scratch tables (tempHeelPrefix).

   **In single-threaded mode**

   The analyses are run in one SQL session and all intermediate results are written to temp tables before finally being combined into the final results tables. Temp tables are dropped once the package is finished running. Single-threaded mode can be invoked by either setting `numThreads = 1` or `scratchDatabaseSchema = "#"`.

   Use the following commands in R:

```
library(Achilles)
connectionDetails <- createConnectionDetails(
  dbms="redshift",
  server="server.com",
  user="secret",
  password='secret',
  port="5439")
```

### Single-threaded mode

```
achilles(connectionDetails,
  cdmDatabaseSchema = "cdm5_inst",
  resultsDatabaseSchema="results",
  vocabDatabaseSchema = "vocab",
  numThreads = 1,
  sourceName = "My Source Name",
  cdmVersion = "5.3.0",
  runHeel = TRUE,
  runCostAnalysis = TRUE)
```

### Multi-threaded mode

```
achilles(connectionDetails,
  cdmDatabaseSchema = "cdm5_inst",
  resultsDatabaseSchema = "results",
  scratchDatabaseSchema = "scratch",
  vocabDatabaseSchema = "vocab",
  numThreads = 10,
  sourceName = "My Source Name",
  cdmVersion = "5.3.0",
  runHeel = TRUE,
  runCostAnalysis = TRUE)
```

The `"cdm5_inst"` cdmDatabaseSchema parameter, `"results"` resultsDatabaseSchema parameter, and `"scratch"` scratchDatabaseSchema parameter are the fully qualified names of the schemas holding the CDM data, targeted for result writing, and holding the intermediate scratch tables, respectively. See the DatabaseConnector package for details on settings the connection details for your database, for example by typing

```
?createConnectionDetails
```

Execution of all Achilles pre-computations may take a long time, particularly in single-threaded mode and with COST analyses enabled. See <extras/notes.md> file to find out how some analyses can be excluded to make the execution faster (excluding cost pre-computations)

The SQL platforms supported by DatabaseConnector and SqlRender are the **only** ones supported here in Achilles as `dbms`. `cdmVersion` can be *ONLY* 5.x (please look at prior commit history for v4 support).

5. To use AchillesWeb to explore the Achilles statistics, you must first export the statistics to a folder JSON files, which can optionally be compressed into one gzipped file for easier transportability.

```
exportToJson(connectionDetails,
  cdmDatabaseSchema = "cdm5_inst",
  resultsDatabaseSchema = "results",
  outputPath = "c:/myPath/AchillesExport",
  compressIntoOneFile = TRUE) # creates gzipped file of all JSON files
```

6. Possible optional additional steps:

   - To see what errors were found (from within R), run: `r      fetchAchillesHeelResults(connectionDetails,re`
   - To see a particular analysis, run: `r      fetchAchillesAnalysisResults(connectionDetails,resultsDataba`
     `= 2)`
   - To join data tables with some lookup (overview files), obtain those using commands below:
   - To get description of analyses, run `getAnalysisDetails()`.
   - To get description of derived measures, run: `r      read.csv(system.file("csv","derived_analysis_detail`
   - Similarly, for overview of rules, run: `r      read.csv(system.file("csv","achilles_rule.csv",package="Ac`
   - Also see notes.md for more information (in the extras folder).

## 1.1 Developers: How to Add or Modify Analyses

Please refer to the README-developers.md file.

## 1.2 License

Achilles is licensed under Apache License 2.0