Package 'Keeper'

May 29, 2024

Type Package

Title An R package to review patient profiles for phenotype validation

Version 0.1.0

Date 2024-05-28

Maintainer Anna <ostropolets@ohdsi.org>

Description An R package to review patient profiles for phenotype validation.

Depends DatabaseConnector (>= 5.0.0),
R (>= 4.0.0)

Imports checkmate,
dplyr,
SqlRender,
english,
stringr

Suggests rmarkdown,
testthat,
knitr,
withr,
Eunomia

License Apache License

RoxygenNote 7.3.1

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

Encoding UTF-8

Language en-US

Contents

createKeeper *Export person level data from OMOP CDM tables for eligible persons in the cohort.*

Description

Use `useAncestor = TRUE` to switch from verbatim string of concept_ids vs ancestors. In latter case, the app will take you concept_ids and include them along with their descendants.

Use `sampleSize` to specify desired number of patients to be selected.

Use `assignNewId = TRUE` to replace person_id with a new sequence.

Explanation of categories:

- instantiated cohort with patients of interest in COHORT table or in another table that has the same fields as COHORT;
- doi: string for disease of interest (ex.: diabetes type I). Hereon, assume a string of concept_ids;
- symptoms: symptoms of disease of interest or alternative/competing diagnoses (those that you want to see to be able to distinguish your doi from another close disease, ex.: polyuria, weight gain or loss, vision disturbances);
- comorbidities: relevant diseases that co-occur with doi or alternative/competing diagnoses (ex.: obesity, metabolic syndrome, pancreatic disorders, pregnancy);
- drugs: drugs, relevant to the disease of interest or those that can be used to treat alternative/competing diagnoses (ex.: insulin, oral glucose lowering drugs);
- diagnosticProcedures: relevant diagnostic procedures (ex.: ultrasound of pancreas);
- measurements: relevant lab tests (ex.: islet cell ab, HbA1C, glucose measurement in blood, insulin ab);
- alternativeDiagnosis: alternative/competing diagnoses (ex.: diabetes type 2, cystic fibrosis, gestational diabetes, renal failure, pancreonecrosis)
- treatmentProcedures: relevant treatment procedures (ex.: operative procedures on pancreas);
- complications: relevant complications (ex.: retinopathy, CKD).

*note: if no suitable concept_ids exists for an input string, input c(0)

Usage

```
createKeeper(
  connectionDetails = NULL,
  connection = NULL,
  cohortDatabaseSchema = NULL,
  cdmDatabaseSchema,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  cohortTable = "cohort",
  cohortDefinitionId,
  cohortName = NULL,
  sampleSize = 20,
  personIds = NULL,
  databaseId,
  assignNewId = FALSE,
  useAncestor = TRUE,
```

```
  doi,
  comorbidities,
  symptoms,
  alternativeDiagnosis,
  drugs,
  diagnosticProcedures,
  measurements,
  treatmentProcedures,
  complications
)
```

Arguments

connectionDetails	An R object of type connectionDetails created using the DatabaseConnector::createConnectio function. Not required of connection is provided.
connection	The connection to the database server created using DatabaseConnector::connect(). Not required if connectionDetails is provided.
cohortDatabaseSchema	The name of the database schema that is the location where the cohort to review is stored.
cdmDatabaseSchema	The name of the database schema that contains the OMOP CDM instance. Requires read permissions to this database. On SQL Server, this should specify both the database and the schema, so for example 'cdm_instance.dbo'.
tempEmulationSchema	Some database platforms like Oracle and Impala do not truly support temp tables. To emulate temp tables, provide a schema with write privileges where temp tables can be created.
cohortTable	The tablename that contains the cohort to review.
cohortDefinitionId	The cohort id to extract records.
cohortName	(optional) Cohort Name
sampleSize	(Optional, default = 20) The number of persons to randomly sample. Ignored, if personId is given.
personIds	(Optional) A vector of personId's to look for in Cohort table and CDM.
databaseId	A short string for identifying the database (e.g. 'Synpuf'). This will be displayed in shiny app to toggle between databases. Should not have space or underscore (_).
assignNewId	(Default = FALSE) Do you want to assign a newId for persons. This will replace the personId in the source with a randomly assigned newId.
useAncestor	keeperOutput: a switch for using concept_ancestor to retrieve relevant terms vs using verbatim strings of codes
doi	keeperOutput: input vector of concept_ids for disease of interest
comorbidities	keeperOutput: input vector of concept_ids for comorbidities associated with the disease of interest (such as smoking or hyperlipidemia for diabetes)
symptoms	keeperOutput: input vector of concept_ids for symptoms associated with the disease of interest (such as weight gain or loss for diabetes)

`alternativeDiagnosis`
: keeperOutput: input vector of concept_ids for competing diagnosis within a month after the index date

`drugs`
: keeperOutput: input vector of concept_ids for drug exposures relevant to the disease of interest, to be used for prior exposures and treatment after the index date. You may input drugs that are used to treat disease of interest and drugs used to treat alternative diagnosis

`diagnosticProcedures`
: keeperOutput: input vector of concept_ids for diagnostic procedures relevant to the condition of interest within a month prior and after the index date

`measurements`
: keeperOutput: input vector of concept_ids for lab tests relevant to the disease of interest within a month prior and after the index date

`treatmentProcedures`
: keeperOutput: input vector of concept_ids for treatment procedures relevant to the disease of interest within a month after the index date

`complications`
: keeperOutput: input vector of concept_ids for complications of the disease of interest within a year after the index date

Value

Output is a data frame with one row per patient, with the following information per patient:

- demographics (age, gender);
- visit_context: information about visits overlapping with the index date (day 0) formatted as the type of visit and its duration;
- observation_period: information about overlapping OBSERVATION_PERIOD formatted as days prior - days after the index date;
- presentation: all records in CONDITION_OCCURRENCE on day 0 with corresponding type and status;
- comorbidities: records in CONDITION_ERA and OBSERVATION that were selected as comorbidities and risk factors within all time prior excluding day 0. The list does not inlcude symptoms, disease of interest and complications;
- symptoms: records in CONDITION_ERA that were selected as symptoms 30 days prior excluding day 0. The list does not include disease of interest and complications. If you want to see symptoms outside of this window, please place them in complications;
- prior_disease: records in CONDITION_ERA that were selected as disease of interest or complications all time prior excluding day 0;
- prior_drugs: records in DRUG_ERA that were selected as drugs of interest all time prior excluding day 0 formatted as day of era start and length of drug era;
- prior_treatment_procedures: records in PROCEDURE_OCCURRENCE that were selected as treatments of interest within all time prior excluding day 0;
- diagnostic_procedures: records in PROCEDURE_OCCURRENCE that were selected as diagnostic procedures within all time prior excluding day 0;
- measurements: records in MEASUREMENT that were selected as measurements (lab tests) of interest within 30 days before and 30 days after day 0 formatted as value and unit (if exists) and assessment compared to the reference range provided in MEASUREMENT table (normal, abnormal high and abnormal low);
- alternative_diagnosis: records in CONDITION_ERA that were selected as alternative (competing) diagnosis within 90 days before and 90 days after day 0. The list does not include disease of interest;

- after_disease: same as prior_disease but after day 0;
- after_drugs: same as prior_drugs but after day 0;
- after_treatment_procedures: same as prior_treatment_procedures but after day 0;
- death: death record any time after day 0.

Examples

```
## Not run:
connectionDetails <- createConnectionDetails(
  dbms = 'postgresql',
  server = 'ohdsi.com',
  port = 5432,
  user = 'me',
  password = 'secure'
)

createKeeper(
  connectionDetails = connectionDetails,
  databaseId = "Synpuf",
  cdmDatabaseSchema = "dbo",
  cohortDatabaseSchema = "results",
  cohortTable = "cohort",
  cohortDefinitionId = 1234,
  cohortName = "DM type I",
  sampleSize = 100,
  assignNewId = TRUE,
  useAncestor = TRUE,
 doi = c(201820,442793,443238,4016045,4065354,45757392, 4051114, 433968, 375545, 29555009,
       4209145, 4034964, 380834, 4299544, 4226354, 4159742, 43530690, 433736, 320128, 4170226,
          40443308, 441267, 4163735, 192963, 85828009),
  symptoms = c(4232487, 4229881),
  comorbidities = c(432867, 436670),
  drugs = c(1730370, 21604490, 21601682, 21601855, 21601462, 21600280, 21602728, 1366773,
            21602689, 21603923, 21603746),
  diagnosticProcedures = c(40756884, 4143852, 2746768, 2746766),
  measurements = c(3034962, 3000483, 3034962, 3000483, 3004501, 3033408, 3005131, 3024629,
                   3031266, 3037110, 3009261, 3022548, 3019210, 3025232, 3033819,
                  3000845, 3002666, 3004077, 3026300, 3014737, 3027198, 3025398, 3010300,
                    3020399, 3007332, 3025673, 3027457, 3010084, 3004410, 3005673),
  alternativeDiagnosis = c(201820,442793,443238,4016045,4065354,45757392, 4051114, 433968,
                     375545, 29555009, 4209145, 4034964, 380834, 4299544, 4226354, 4159742,
                      43530690, 433736, 320128, 4170226, 40443308, 441267, 4163735, 192963,
                             85828009),
   treatmentProcedures = c(0),
  complications =  c(201820,442793,443238,4016045,4065354,45757392, 4051114, 433968, 375545,
                  29555009, 4209145, 4034964, 380834, 4299544, 4226354, 4159742, 43530690,
                    433736, 320128, 4170226, 40443308, 441267, 4163735, 192963, 85828009)
)

## End(Not run)
```

`createPrompt` *Create the main prompt based on a Keeper output row.*

Description

Create the main prompt based on a Keeper output row.

Usage

```
createPrompt(settings, diseaseName, keeperRow)
```

Arguments

settings	A settings object as created using `createPromptSettings()`.
diseaseName	The name of the disease to use in the prompt.
keeperRow	A single row from the output of `createKeeper()`.

Value

A character string containing the main prompt.

---

createPromptSettings *Create settings for generating prompts*

Description

Create settings for generating prompts

Usage

```
createPromptSettings(
  writeNarrative = TRUE,
  testingReminder = TRUE,
  uncertaintyInstructions = TRUE,
  discussEvidence = TRUE,
  provideExamples = FALSE,
  maxParts = 100,
  maxDays = 5
)
```

Arguments

writeNarrative	Ask the LLM to write a clinical narrative matching the provided data?
testingReminder	Remind the LLM that a diagnosis can be recorded just to justify a test, and therefore by itself is not sufficient evidence?
uncertaintyInstructions	Provide instructions to the LLM on how to deal with uncertainty?
discussEvidence	Prompt the LLM to first discuss evidence in favor and against the disease of interest?
provideExamples	Provide examples? (few-shot prompting)

maxParts	How many parts can a category have? For example, if maxParts = 100 and there are more than 100 measurements, a random sample of 100 will be taken. Set to 0 if there is no maximum.
maxDays	How many days can a single code have? For example, if maxDays = 5 and there is a measurement code that appears on more than 5 days, a random sample of 5 days will be taken. Set to 0 if there is no maximum.

Value

A settings object, to be used in createSystemPrompt() and createPrompt().

---

createSystemPrompt *Create a system prompt for a LLM*

Description

Create a system prompt for a LLM

Usage

```
createSystemPrompt(settings, diseaseName)
```

Arguments

settings	A settings object as created using createPromptSettings().
diseaseName	The name of the disease to use in the prompt.

Value

A character string with the system prompt.

---

parseLlmResponse *Parse the response of a LLM*

Description

Parse the response of a LLM

Usage

```
parseLlmResponse(response, noMatchIsDontKnow = TRUE)
```

Arguments

response	The response of a LLM to the system prompt with prompt generated by createSystemPrompt() and createPrompt(), respectively.
noMatchIsDontKnow	If the response doesn't fit any predefined pattern, should we return "I don't know"?

Value

Returns a character string with one of the following values:

- "yes": Yes, the patient has the disease.
- "no": No, the patient does not have the disease.
- "I don't know": The LLM cannot decide whether the patient has the disease.
- NA: There was a problem parsing the LLM's response.

Examples

```
parseLlmResponse("Summary: yes")
parseLlmResponse("Summary: It is unclear whether the patient has the disease.")
```

Index