

Package ‘Aphrodite’

June 10, 2015

Type Package

Title Automated PHenotype Routine for Observational Definition Identification Training and Evaluation (APHRODITE) - Phenotype building tool using Fuzzy labels

Version 1.0

Date 2015-06-09

Author Juan M. Banda [aut, cre],
Marc A. Suchard [aut],
Martijn J. Schuemie [aut]

Maintainer Juan M. Banda <jmbanda@stanford.edu>

Description Aphrodite uses noisy class labels to create silver standard training corpora to construct phenotype models in conjunction with expert knowledge codified in existing ontologies and a comprehensive representation of the patient clinical record to learn phenotype models.

License Apache License 2.0 | file LICENSE

Depends R (>= 3.1.0)

Imports DatabaseConnector,
SqlRender,
plyr,
caret,
pROC,
devtools,
knitr,
testthat

Suggests ggplot2

VignetteBuilder knitr

R topics documented:

buildFeatureVector	2
buildKeywordList	2
buildModel	3
executeSQL	4
getdPatientCohort	5
getPatientData	6
getPatientDataCases	7
Index	8

`buildFeatureVector` *This function builds a feature vector using raw patient data*

Description

This function builds a feature vector using raw patient data. Returns a patient feature vector (divided by feature sets).

Usage

```
buildFeatureVector(flags, casesS, controlsS)
```

Arguments

<code>flags</code>	The R dataframe that contains all feature/model flags specified in settings.R.
<code>casesS</code>	Dataframe containing the raw patient data.
<code>controlsS</code>	(OPTIONAL) Dataframe containing the raw patient data.

Details

This function flattens the patient feature data (per feature set) into a feature vector that will be used as input for caret. This function can optionally flat two sources of patient data (cases and controls)

Value

An object containing the flattened feature vectors for all given feature sets.

Examples

```
## Not run:

fv_all<-buildFeatureVector(flag, dataFcases,dataFcontrols)

#OR

fv_cases<-buildFeatureVector(flag, dataFcases)

## End(Not run)
```

`buildKeywordList` *This function generates keyword and ignore lists based on the expansion of concepts.*

Description

Given any given concept_id or string of text this function generates keyword and ignore lists based on the expansion of concepts (looking at their synonyms).

Usage

```
buildKeywordList(connection, aphroditeConceptName, schema, dbms)
```

Arguments

connection	The connection to the database server.
aphroditeConceptName	The string of text / concept name to use.
schema	The database schema being used.
dbms	The target DBMS for SQL to be rendered in.

Details

Takes the aphroditeConceptName looks for synonyms and builds a list of related concepts using the vocabulary hierarchies

Value

A list with two elements: a list of positive keywords found (keywordlist_ALL), and a list of ignore keywords (ignorelist_ALL)

Examples

```
## Not run:

wordLists <- buildKeywordList(conn, aphrodite_concept_name, cdmSchema, dbms)

## End (Not run)
```

buildModel	<i>This function builds a model for the specified feature vector using cases and controls for a certain outcomeName</i>
------------	---

Description

This function builds a model for the specified feature vector using cases and controls for a certain outcomeName. Returns a caret trained model.

Usage

```
buildModel(flags, cases_pids, controls_pids, featureVector, outcomeNameS)
```

Arguments

flags	The R dataframe that contains all feature/model flags specified in settings.R.
cases_pids	List of patient_id's considered cases (for labeling purposes)
controls_pids	List of patient_id's considered controls (for labeling purposes)
featureVector	Flattened feature vector returned by buildFeatureVector function.
outcomeNameS	String description of the outcome for which the model is being built

Details

This function builds a model for the specified outcomeName. The model is specified in the flags dataframe (currently only supports LASSO). The cases_pids and controld_pids are patient_id's used for the labeling of the testing and training sets.

Value

An transferable caret Model object

Examples

```
## Not run:

model_predictors <- buildModel(flag, cases, controls, fv_all, outcomeName)

## End (Not run)
```

executeSQL

This function executes one single SQL statement

Description

This function renders, translates and executes one single SQL statement that produces a result

Usage

```
executeSQL(connection, schema, query, targetDBMS)
```

Arguments

connection	The connection to the database server.
schema	The database schema being used.
query	The SQL statement to retrieve the data.
targetDBMS	The target DBMS for SQL to be rendered in.

Details

Renders, translates, and executes a single SQL statement that is expeting to produce and result.

Value

An object containing the data.

Examples

```
## Not run:

library("SqlRender")
library("DatabaseConnector")
library("Aphordite")
connectionDetails <- createConnectionDetails(dbms="mysql", server="localhost",
  user="root", password="blah", schema="cdm_v5")
conn <- connect(connectionDetails)

concept_of_interest <- executeSQL(connection, schema, paste("SELECT concept_id,
  concept_name FROM @cdmSchema.concept WHERE lower(concept_name) =
  lower('myocardial infarction') AND standard_concept = 'S' AND
  invalid_reason IS NULL AND domain_id = 'Condition';", sep = ""), dbms)

dbDisconnect(conn)

## End(Not run)
```

getdPatientCohort	<i>This function builds a patient cohort (and controls) based on a concept list</i>
-------------------	---

Description

This function will build a patient cohort with its respective controls using an inclusion concept_id list as well as an exclusion concept_id list. The user specifies the number of both cases and controls for his cohort.

Usage

```
getdPatientCohort(connection, dbms, includeConceptlist, excludeConceptlist,
  schema, cohortSize, controlSize)
```

Arguments

connection	The connection to the database server.
dbms	The target DBMS for SQL to be rendered in.
includeConceptlist	The list of concept_id's used to build the cohort.
excludeConceptlist	The list of concept_id's used as exclusion criteria for the cohort.
schema	The database schema being used.
cohortSize	The number of desired patients to appear in the cohort.
controlSize	The number of desired patients to be in the control group.

Details

This function takes the lists of include and exclude concept_ids and finds all patients that satisfy this characteristics from the Observation and Condition_occurrence tables in CDM V5.

Value

A list of dataframes containing both cases and control patient_id's.

Examples

```
## Not run:

casesANDcontrolspatient_ids_df<- getdPatientCohort(conn, dbms,
  as.character(keywordList_FF$V3), as.character(ignoreList_FF$V3),
  cdmSchema,nCases,nControls)
if (nCases > nrow(casesANDcontrolspatient_ids_df[[1]])) {
  message("Not enough patients to get the number of cases specified")
  stop
} else {
  if (nCases > nrow(casesANDcontrolspatient_ids_df[[2]])) {
    message("Not enough patients to get the number of controls specified")
    stop
  }
}

## End(Not run)
```

getPatientData	<i>This function fetches all the patient data (generic)</i>
----------------	---

Description

This function fetches all the patient data (generic). Returns raw patient data.

Usage

```
getPatientData(connection, dbms, patient_ids, flags, schema)
```

Arguments

connection	The connection to the database server.
dbms	The target DBMS for SQL to be rendered in.
patient_ids	The list of case patient id's to extract data from.
flags	The R dataframe that contains all feature/model flags specified in settings.R.
schema	The database schema being used.

Details

Based on the groups of feature sets determined in the flags variable, this function will fetch patient data. The function returns all patient information

Value

An object containing the raw feature sets for the patient data.

Examples

```
## Not run:

dataFcontrols <- getPatientData(conn, dbms, controls, flag , cdmSchema)

## End(Not run)
```

```
getPatientDataCases
```

This function fetches all the patient data (non-generic) designed to work when building a model

Description

This function fetches all the patient data (non-generic) designed to work when building a model. Returns raw patient data.

Usage

```
getPatientDataCases(connection, dbms, patient_ids, keywords, ignores, flags,
  schema)
```

Arguments

connection	The connection to the database server.
dbms	The target DBMS for SQL to be rendered in.
patient_ids	The list of case patient id's to extract data from.
keywords	The list of concept_id's used to build the cohort.
ignores	The list of concept_id's ignored when building the cohort.
flags	The R dataframe that contains all feature/model flags specified in settings.R.
schema	The database schema being used.

Details

Based on the groups of feature sets determined in the flags variable, this function will fetch patient data. The function determines the first mention of the keywords and selects that date to start the data extraction of the remaining patient information

Value

An object containing the raw feature sets for the patient data.

Examples

```
## Not run:

dataFcases <-getPatientDataCases(conn, dbms, cases, as.character(keywordList_FF$V3),
  flag , cdmSchema)

## End(Not run)
```

Index

buildFeatureVector, [2](#)
buildKeywordList, [2](#)
buildModel, [3](#)

executeSQL, [4](#)

getdPatientCohort, [5](#)
getPatientData, [6](#)
getPatientDataCases, [7](#)