Package ‘EHRtemporalVariability’

May 25, 2020

Type Package

Title Delineating Temporal Dataset Shifts in Electronic Health Records

Version 1.1.1

Date 2020-05-25

Encoding UTF-8

Description Functions to delineate temporal dataset shifts in Electronic Health Records through the projection and visualization of dissimilarities among data temporal batches. This is done through the estimation of data statistical distributions over time and their projection in non-parametric statistical manifolds, uncovering the patterns of the data latent temporal variability. 'EHRtemporalVariability' is particularly suitable for multi-modal data and categorical variables with a high number of values, common features of biomedical data where traditional statistical process control or time-series methods may not be appropriate. 'EHRtemporalVariability' allows you to explore and identify dataset shifts through visual analytics formats such as Data Temporal heatmaps and Information Geometric Temporal (IGT) plots. An additional 'EHRtemporalVariability' Shiny app can be used to load and explore the package results and even to allow the use of these functions to those users non-experienced in R coding. Preprint published in medRxiv (Sáez et al. 2020) <doi:10.1101/2020.04.07.20056564>.

Depends R (>= 3.3.0), dplyr

License Apache License 2.0 | file LICENSE

LazyData true

Imports plotly, zoo, xts, lubridate, RColorBrewer, viridis, scales, methods, MASS

Suggests knitr, rmarkdown, devtools, BiocStyle, dbscan

VignetteBuilder knitr

NeedsCompilation no

Maintainer Carlos Sáez <carsaesi@upv.es>

RoxygenNote 7.1.0

URL http://github.com/hms-dbmi/EHRtemporalVariability

Author Carlos Sáez [aut, cre],
Alba Gutiérrez-Sacristán [aut],
Isaac Kohane [aut],
Juan M García-Gómez [aut],
Paul Avillach [aut],
Biomedical Data Science Lab, Universitat Politècnica de València
(Spain) [cph],
Department of Biomedical Informatics, Harvard Medical School [cph]

Repository CRAN

Date/Publication 2020-05-25 14:10:02 UTC

R topics documented:

`DataTemporalMap-class` *Class DataTemporalMap*

Description

Class `DataTemporalMap` object contains the statistical distributions of data estimated at a specific time period. Both relative and absolute frequencies are included.

Details

Objects of this class are generated automatically by the `estimateDataTemporalMap` function, but its construction and extension is open towards fostering its use through external methods. E.g., one may use additional probability distribution estimation methods, or even construct compatible DataTemporalMaps for other unstructured data such as images or free text.

Value

A `DataTemporalMap` object.

Slots

- `probabilityMap` v-by-d numerical `matrix` representing the probability distribution temporal map (relative frequency).
- `countsMap` v-by-d numerical `matrix` representing the counts temporal map (absolute frequency).
- `dates` d-dimensional `Date` array of the temporal batches.
- `support` v-by-1 numerical or character `matrix` representing the support (the value at each bin) of probabilityMap and countsMap.
- `variableName` name of the variable (character).
- `variableType` type of the variable (character) among "numeric", "character", "Date" and "factor".
- `period` batching period among "week", "month" and "year".

Examples

```
# Generation through estimateDataTemporalMap function:
dataset <- read.csv2(system.file("extdata",
                                  "nhdsSubset.csv",
                                  package="EHRtemporalVariability"),
                     sep  = ",",
                     header = TRUE,
                     na.strings = "",
                     colClasses = c( "character", "numeric", "factor",
                                     "numeric" , rep( "factor", 22 ) ) )

datasetFormatted <- EHRtemporalVariability::formatDate(
                     input         = dataset,
                     dateColumn    = "date",
                     dateFormat = "%y/%m")

probMaps <- estimateDataTemporalMap(data = datasetFormatted,
                     dateColumnName = "date",
                     period         = "month")

class( probMaps[[1]] )

# Manual generation:
countsMatrix <- matrix(sample.int(25, size = 12*10, replace = TRUE), nrow = 12, ncol = 10)
probabilityMatrix <- sweep(countsMatrix,1,rowSums(countsMatrix),"/")
dates <- seq(Sys.Date(),(Sys.Date()+30*11),30)
x <- new('DataTemporalMap', probabilityMap = probabilityMatrix,
                    countsMap = countsMatrix, dates = dates, support = data.frame(1:10),
                   variableName = "example", variableType = "numeric", period = "month")
plotDataTemporalMap(x)
```

`estimateDataTemporalMap` *Estimates* `DataTemporalMap` *objects from raw data*

Description

Estimates a `DataTemporalMap` from a `data.frame` containing individuals in rows and the variables in columns, being one of these columns the analysis date (typically the acquisition date). Will return a `DataTemporalMap` object or a `list` of `DataTemporalMap` objects depending on the number of analysis variables.

Usage

```
estimateDataTemporalMap(
  data = NULL,
  dateColumnName = NULL,
  period = "month",
  startDate = NULL,
  endDate = NULL,
  supports = NULL,
  numericVariablesBins = 100,
  numericSmoothing = TRUE,
  dateGapsSmoothing = FALSE,
  verbose = FALSE
)
```

Arguments

`data` a `data.frame` containing as many rows as individuals, and as many columns as the analysis variables plus the individual acquisition date.

`dateColumnName` a string indicating the name of the column in `data` containing the analysis date variable.

`period` the period at which to batch data for the analysis from "week", "month" and "year", with "month" as default.

`startDate` a Date object indicating the date at which to start the analysis, in case of being different from the first chronological date in the date column (the default).

`endDate` a Date object indicating the date at which to end the analysis, in case of being different from the last chronological date in the date column (the default).

`supports` a List of objects containing the support of the data distributions for each variable, in classes `numeric`, `integer`, `character`, or `factor` (accordingly to the variable type), and where the name of the list element must correspond to the column name of its variable. If not provided it is automatically estimated from data.

`numericVariablesBins`
the number of bins at which to define the frequency/density histogram for numerical variables when their support is not provided, 100 as default.

numericSmoothing
a logical value indicating whether a Kernel Density Estimation smoothing (Gaussian kernel, default bandwidth) is to be applied on numerical variables (the default) or a traditional histogram instead. See ?density for further details.

dateGapsSmoothing
a logical value indicating whether a linear smoothing is applied to those time batches without data, by default gaps are filled with NAs.

verbose
By default `FALSE`. Change it to `TRUE` to get an on-time log from the function.

Value

A `DataTemporalMap` object.

Examples

```
#Load the file
dataset <- read.csv2(system.file("extdata",
                                   "nhdsSubset.csv",
                                   package="EHRtemporalVariability"),
                     sep  = ",",
                     header = TRUE,
                     na.strings = "",
                     colClasses = c( "character", "numeric", "factor",
                                     "numeric" , rep( "factor", 22 ) ) )
#Format the date
datasetFormatted <- EHRtemporalVariability::formatDate( input         = dataset,
                                            dateColumn    = "date",
                                            dateFormat    = "%y/%m")

#Apply the estimateDataTemporalMap
probMaps <- estimateDataTemporalMap( data           = datasetFormatted,
                                     dateColumnName = "date",
                                     period         = "month")
## Not run:

For a larger example download the following .csv dataset and continue the steps as above:

gitHubUrl  <- 'http://github.com/'
gitHubPath <- 'hms-dbmi/EHRtemporalVariability-DataExamples/'
gitHubFile <- 'raw/master/nhdsSubset.csv'
inputFile  <-  paste0(gitHubUrl, gitHubPath, gitHubFile)

dataset <- read.csv2( inputFile,
                     sep  = ",",
                     header = TRUE,
                     na.strings = "",
                     colClasses = c( "character", "numeric", "factor",
                                     "numeric" , rep( "factor", 22 ) ) )

## End(Not run)
```

estimateIGTProjection *Estimates an Information Geometric Temporal plot projection*

Description

Estimates an `IGTProjection` object from a `DataTemporalMap` object.

Usage

```
estimateIGTProjection(
  dataTemporalMap,
  dimensions = 3,
  startDate = NULL,
  endDate = NULL,
  embeddingType = "classicalmds"
)

## S4 method for signature 'DataTemporalMap'
estimateIGTProjection(
  dataTemporalMap,
  dimensions = 3,
  startDate = NULL,
  endDate = NULL,
  embeddingType = "classicalmds"
)
```

Arguments

`dataTemporalMap`
of class `DataTemporalMap` object.

`dimensions` `numeric` integer value indicating the number of dimensions for the projection.

`startDate` a Date object indicating the date at which to start the analysis, in case of being different from the first chronological date in the date column (the default).

`endDate` a Date object indicating the date at which to end the analysis, in case of being different from the last chronological date in the date column (the default).

`embeddingType` the type of embedding to apply to the dissimilarity matrix of time batches in order to obtain the non-parametric Statistical Manifold, from "classicalmds" and "nonmetricmds", with "classicalmds" as default. "classicalmds" uses the base R stats::cmdscale function, while "nonmetricmds" uses the MASS:isoMDS function. The returned stress format will depend on the selected embedding type: "classicalmds" returns 1-GOF as returned by stats::cmdscale function, "nonmetricmds" returns the final stress in percent, as returned by the MASS::isoMDS function

Value

An IGTProjection object containing the projected coordinates of each temporal batch in the embedded non-parametric Statistical Manifold, as well as the embedding stress according to the embeddingType.

Examples

```
load(system.file("extdata",
                 "variabilityDemoNHDSdiagcode1-phewascode.RData",
                  package="EHRtemporalVariability"))
igtProj <- estimateIGTProjection( dataTemporalMap = probMaps$`diagcode1-phewascode`,
dimensions      = 3,
startDate       = "2000-01-01",
endDate         = "2010-12-31")

## Not run:

# For additional and larger examples download the following .Rdata file:

gitHubUrl  <- 'http://github.com/'
gitHubPath <- 'hms-dbmi/EHRtemporalVariability-DataExamples/'
gitHubFile <- 'raw/master/variabilityDemoNHDS.RData'
inputFile  <-  paste0(gitHubUrl, gitHubPath, gitHubFile)

load(url(inputFile))
igtProj <- estimateIGTProjection( dataTemporalMap = probMaps[[1]],
dimensions      = 3,
startDate       = "2000-01-01",
endDate         = "2010-12-31")

## End(Not run)
```

---

estimateIGTTrajectory *Estimates a trajectory of the information temporal evolution in a IGT projection by fitting a cubic smoothing spline*

Description

Estimates a DataTemporalMap from a data.frame containing individuals in rows and the variables in columns, being one of these columns the analysis date (typically the acquisition date). Will return a DataTemporalMap object or a list of DataTemporalMap objects depending on the number of analysis variables.

Usage

```
estimateIGTTrajectory(igtProjection, nPoints = NULL)
```

Arguments

igtProjection	of class IGTProjection.
nPoints	the number of points to fit within the IGT projection range. By default 10x the number of time batches, what shows a high resolution trajectory.

Value

A list containing a data.frame of the estimated trajectory points, the estimated date for each point, and the fitted trajectory function as smooth.spline objects.

Examples

```
load(system.file("extdata",
                 "variabilityDemoNHDSdiagcode1-phewascode.RData",
                  package="EHRtemporalVariability"))

igtTrajectory <- estimateIGTTrajectory( igtProjection   =  igtProjs[[1]] )
igtTrajectory$points
```

formatDate *Function to transform dates into "Date" R format*

Description

Given a data.frame object with a column of dates in 'character' format, it generates a new data.frame object with the dates transformed into "Date" R format.

Usage

```
formatDate(input, dateColumn, dateFormat = "%y/%m/%d", verbose = FALSE)
```

Arguments

input	A data.frame object with at least one column of dates.
dateColumn	The name of the column containing the date.
dateFormat	By default '%y/%m/%d'. Change it to the specific structure of your date format.
verbose	By default FALSE. Change it to TRUE to get an on-time log from the function.

Value

An object of class data.frame with the date column transform into 'Date' R class.

Examples

```
dataset <- read.csv2(system.file("extdata",
                                 "nhdsSubset.csv",
                                 package="EHRtemporalVariability"),
                     sep  = ",",
                     header = TRUE,
                     na.strings = "",
                     colClasses = c( "character", "numeric", "factor",
                                     "numeric" , rep( "factor", 22 ) ) )

datasetFormatted <- formatDate(
              input         = dataset,
              dateColumn    = "date",
              dateFormat    = "%y/%m",
             )
```

`icd9toPheWAS` *Function to transform ICD9-CM codification into PheWAS code*

Description

Given a `data.frame` object with a column of ICD9-CM codes, it generates a new `data.frame` object with the ICD9-CM codes transformed into PheWAS codes.

Usage

```
icd9toPheWAS(
  data,
  icd9ColumnName,
  missingValues = "NA",
  phecodeDescription = FALSE,
  statistics = FALSE,
  replaceColumn = TRUE,
  verbose = FALSE
)
```

Arguments

`data`	A `data.frame` object with at least one column of ICD9-CM codes that one to be transformed into a PheWAS code.
`icd9ColumnName`	The name of the column containing the ICD9-CM.
`missingValues`	The value used to determine missing values in the data.frame.
`phecodeDescription`	By default `FALSE`. Change it to `TRUE` to map to the PheWAS code description instead to the PheWAS numeric code.

`statistics` By default `FALSE`. Change it to `TRUE` to show the summary of the mapping like the percentage of initial ICD9-CM codes mapped to PheWAS code.

`replaceColumn` By default `TRUE`. Change it to `FALSE` in order to create a new column with the PheWAS code maintaining the ICD9-CM code.

`verbose` By default `FALSE`. Change it to `TRUE` to get an on-time log from the function.

Value

An object of class `data.frame` with the ICD9-CM column transform into PheWAS codes.

Examples

```
dataset <- read.csv2(system.file("extdata",
                                   "nhdsSubset.csv",
                                   package="EHRtemporalVariability"),
                     sep  = ",",
                     header = TRUE,
                     na.strings = "",
                     colClasses = c( "character", "numeric", "factor",
                                     "numeric" , rep( "factor", 22 ) ) )

datasetPheWAS <- icd9toPheWAS( data           = dataset,
                              icd9ColumnName  = "diagcode1",
                              missingValues   = "N/A",
                              replaceColumn   = TRUE,
                              statistics      = TRUE
                              )
```

plotDataTemporalMap *Data Temporal heatmap*

Description

Plots a Data Temporal heatmap from an `DataTemporalMap` object.

Usage

```
plotDataTemporalMap(
  dataTemporalMap,
  absolute = FALSE,
  startValue = 1,
  endValue = ncol(dataTemporalMap@probabilityMap),
  startDate = min(dataTemporalMap@dates),
  endDate = max(dataTemporalMap@dates),
  sortingMethod = "frequency",
  colorPalette = "Spectral"
)
```

```
## S4 method for signature 'DataTemporalMap'
plotDataTemporalMap(
  dataTemporalMap,
  absolute = FALSE,
  startValue = 1,
  endValue = ncol(dataTemporalMap@probabilityMap),
  startDate = min(dataTemporalMap@dates),
  endDate = max(dataTemporalMap@dates),
  sortingMethod = "frequency",
  colorPalette = "Spectral"
)
```

Arguments

`dataTemporalMap` of class `DataTemporalMap`

`absolute` indicates if the heatmap frequency values are absolute or relative. By default `FALSE`.

`startValue` indicates the first value to display in the heatmap. By default 1.

`endValue` indicates the last value to display in the heatmap. By default the last value of the `DataTemporalMap` object.

`startDate` a Date object indicating the first date to be displayed in the heatmap. By default the first date of the `DataTemporalMap` object.

`endDate` a Date object indicating the last date to be displayed in the heatmap. By default the last date of the `DataTemporalMap` object.

`sortingMethod` the method to sort data in the Y axis of the heatmap from "frequency" and "alphabetical", with "frequency" as default.

`colorPalette` color palette to be used. The default "Spectral" palette shows a color temperature scheme from blue, through yellow, to red (see "Spectral" palette in RColorBrewer package). The four remaining options are better suited for those with colorblindness, including "Viridis", "Magma", and their reversed versions "Viridis-reversed" and "Magma-reversed" (see "Viridis" and "Magma" palettes in the Viridis package).

Value

A plot object based on the `plotly` package.

Examples

```
load(system.file("extdata",
                 "variabilityDemoNHDSdiagcode1-phewascode.RData",
                  package="EHRtemporalVariability"))

p <- plotDataTemporalMap(dataTemporalMap =  probMaps[[1]],
                    colorPalette    = "Spectral",
                    startValue = 2,
                    endValue = 40)
```

```
p

## Not run:

# For additional and larger examples download the following .Rdata file:

gitHubUrl  <- 'http://github.com/'
gitHubPath <- 'hms-dbmi/EHRtemporalVariability-DataExamples/'
gitHubFile <- 'raw/master/variabilityDemoNHDS.RData'
inputFile  <-  paste0(gitHubUrl, gitHubPath, gitHubFile)

load(url(inputFile))
plotDataTemporalMap(probMaps$`diagcode1-phewascode`, startValue = 2, endValue = 40)

## End(Not run)
```

plotIGTProjection *Information Geometric Temporal plot*

Description

Plots an interactive Information Geometric Temporal (IGT) plot from an `IGTProjection` object. An IGT plot visualizes the variability among time batches in a data repository in a 2D or 3D plot. Time batches are positioned as points where the distance between them represents the probabilistic distance between their distributions (currently Jensen-Shannon distance, more distances will be supported in the future). To track the temporal evolution, temporal batches are labeled to show their date and colored according to their season or period, according to the analysis period, as follows. If period=="year" the label is "yy" (2 digit year) and the color is according to year. If period=="month" the label is "yym" (yy + abbreviated month*) and the color is according to the season (yearly). If period=="week" the label is "yymmw" (yym + ISO week number in 1-2 digit) and the color is according to the season (yearly). An estimated smoothed trajectory of the information evolution over time can be shown using the optional "trajectory" parameter. *Month abbreviations: {’J’, ’F’, ’M’, ’A’, ’m’, ’j’, ’x’, ’a’, ’S’, ’O’, ’N’, ’D’}.

Usage

```
plotIGTProjection(
  igtProjection,
  dimensions = 3,
  startDate = min(igtProjection@dataTemporalMap@dates),
  endDate = max(igtProjection@dataTemporalMap@dates),
  colorPalette = ”Spectral”,
  trajectory = FALSE
)

## S4 method for signature 'IGTProjection'
plotIGTProjection(
  igtProjection,
  dimensions = 3,
  startDate = min(igtProjection@dataTemporalMap@dates),
  endDate = max(igtProjection@dataTemporalMap@dates),
  colorPalette = "Spectral",
  trajectory = FALSE
)
```

Arguments

`igtProjection`	of class `IGTProjection`
`dimensions`	number of dimensions of the plot, 2 or 3 (3 by default)
`startDate`	a Date object indicating the first date to be displayed in the IGT plot. By default the first date of the `IGTProjection` object.
`endDate`	a Date object indicating the last date to be displayed in the IGT plot By default the last date of the `IGTProjection` object.
`colorPalette`	color palette to be used. The default "Spectral" palette shows a color temperature scheme from blue, through yellow, to red (see "Spectral" palette in RColorBrewer package). The four remaining options are better suited for those with colorblindness, including "Viridis", "Magma", and their reversed versions "Viridis-reversed" and "Magma-reversed" (see "Viridis" and "Magma" palettes in the Viridis package).
`trajectory`	whether to show an estimated trajectory of the information evolution over time. By default `FALSE`.

Details

Note that since the projection is based on multidimensional scaling, a 2 dimensional projection entails a loss of information compared to a 3 dimensional projection. E.g., periodic variability components such as seasonal effect can be hindered by an abrupt change or a general trend.

Value

A plot object based on the `plotly` package.

Examples

```
load(system.file("extdata",
                 "variabilityDemoNHDSdiagcode1-phewascode.RData",
                  package="EHRtemporalVariability"))

p <- plotIGTProjection( igtProjection   =  igtProjs[[1]],
                        colorPalette    = "Spectral",
                        dimensions      = 2)
p

## Not run:

# For additional and larger examples download the following .Rdata file:
```

```
gitHubUrl  <- 'http://github.com/'
gitHubPath <- 'hms-dbmi/EHRtemporalVariability-DataExamples/'
gitHubFile <- 'raw/master/variabilityDemoNHDS.RData'
inputFile  <-  paste0(gitHubUrl, gitHubPath, gitHubFile)

load(url(inputFile))
plotIGTProjection(igtProjs$`diagcode1-phewascode`, dimensions = 3)

## End(Not run)
```

trimDataTemporalMap *Trims a* DataTemporalMap

Description

Trims a DataTemporalMap object between an start and end date. If one is not specified it takes as default the first/last chronological date in the input DataTemporalMap.

Usage

```
trimDataTemporalMap(
  dataTemporalMap,
  startDate = min(dataTemporalMap@dates),
  endDate = max(dataTemporalMap@dates)
)

## S4 method for signature 'DataTemporalMap'
trimDataTemporalMap(
  dataTemporalMap,
  startDate = min(dataTemporalMap@dates),
  endDate = max(dataTemporalMap@dates)
)
```

Arguments

dataTemporalMap
of class DataTemporalMap.

startDate Date indicating the start date to trim from.

endDate Date indicating the end date to trim to.

Value

A DataTemporalMap object between the specified dates.

Examples

```
load(system.file("extdata",
                 "variabilityDemoNHDSdiagcode1-phewascode.RData",
                  package="EHRtemporalVariability"))

probMapTrimmed <- trimDataTemporalMap(
                         dataTemporalMap = probMaps[[1]],
                         startDate       = "2005-01-01",
                         endDate         = "2008-12-01"
)
```

Index