

Package ‘EHRtemporalVariability’

May 20, 2020

Type Package

Title Delineating Temporal Dataset Shifts in Electronic Health Records

Version 1.1.0

Date 2020-05-19

Encoding UTF-8

Description Contains 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 suited to 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 suitable. '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, shiny, zoo, xts, lubridate, RColorBrewer, viridis, scales, methods, MASS, dbscan

Suggests knitr, rmarkdown, devtools, BiocStyle

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]

R topics documented:

DataTemporalMap-class	2
estimateDataTemporalMap	3
estimateIGTProjection	5
estimateIGTTrajectory	7
formatDate	7
icd9toPheWAS	8
plotDataTemporalMap	9
plotIGTProjection	11
trimDataTemporalMap	13

Index	14
--------------	-----------

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

<code>data</code>	a <code>data.frame</code> containing as many rows as individuals, and as many columns as the analysis variables plus the individual acquisition date.
<code>dateColumnName</code>	a string indicating the name of the column in <code>data</code> containing the analysis date variable.
<code>period</code>	the period at which to batch data for the analysis from "week", "month" and "year", with "month" as default.
<code>startDate</code>	a <code>Date</code> 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).
<code>endDate</code>	a <code>Date</code> 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).
<code>supports</code>	a List of objects containing the support of the data distributions for each variable, in classes <code>numeric</code> , <code>integer</code> , <code>character</code> , or <code>factor</code> (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 <code>data</code> .
<code>numericVariablesBins</code>	the number of bins at which to define the frequency/density histogram for numerical variables when their support is not provided, 100 as default.
<code>numericSmoothing</code>	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 <code>?density</code> for further details.
<code>dateGapsSmoothing</code>	a logical value indicating whether a linear smoothing is applied to those time batches without data, by default gaps are filled with NAs.
<code>verbose</code>	By default <code>FALSE</code> . Change it to <code>TRUE</code> 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

<code>dataTemporalMap</code>	of class <code>DataTemporalMap</code> object.
<code>dimensions</code>	numeric integer value indicating the number of dimensions for the projection.
<code>startDate</code>	a <code>Date</code> 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).
<code>endDate</code>	a <code>Date</code> 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).
<code>embeddingType</code>	the type of embedding to apply to the dissimilarity matrix of time batches in order to obtain the non-parametric Statistical Manifold, from "classicalmids" and "nonmetricmids", with "classicalmids" as default. "classicalmids" uses the base R <code>stats::cmdscale</code> function, while "nonmetricmids" uses the MASS::isoMDS function. The returned stress format will depend on the selected embedding type: "classicalmids" returns 1-GOF as returned by <code>stats::cmdscale</code> function, "nonmetricmids" 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 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

<code>input</code>	A <code>data.frame</code> object with at least one column of dates.
<code>dateColumn</code>	The name of the column containing the date.
<code>dateFormat</code>	By default <code>'%y/%m/%d'</code> . Change it to the specific structure of your date format.
<code>verbose</code>	By default <code>FALSE</code> . Change it to <code>TRUE</code> 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

<code>data</code>	A <code>data.frame</code> object with at least one column of ICD9-CM codes that one to be transformed into a PheWAS code.
<code>icd9ColumnName</code>	The name of the column containing the ICD9-CM.
<code>missingValues</code>	The value used to determine missing values in the <code>data.frame</code> .
<code>phecodeDescription</code>	By default <code>FALSE</code> . Change it to <code>TRUE</code> to map to the PheWAS code description instead to the PheWAS numeric code.
<code>statistics</code>	By default <code>FALSE</code> . Change it to <code>TRUE</code> to show the summary of the mapping like the percentage of initial ICD9-CM codes mapped to PheWAS code.
<code>replaceColumn</code>	By default <code>TRUE</code> . Change it to <code>FALSE</code> in order to create a new column with the PheWAS code maintaining the ICD9-CM code.
<code>verbose</code>	By default <code>FALSE</code> . Change it to <code>TRUE</code> 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

<code>dataTemporalMap</code>	of class <code>DataTemporalMap</code>
<code>absolute</code>	indicates if the heatmap frequency values are absolute or relative. By default <code>FALSE</code> .
<code>startValue</code>	indicates the first value to display in the heatmap. By default 1.
<code>endValue</code>	indicates the last value to display in the heatmap. By default the last value of the <code>DataTemporalMap</code> object.
<code>startDate</code>	a Date object indicating the first date to be displayed in the heatmap. By default the first date of the <code>DataTemporalMap</code> object.
<code>endDate</code>	a Date object indicating the last date to be displayed in the heatmap. By default the last date of the <code>DataTemporalMap</code> object.
<code>sortingMethod</code>	the method to sort data in the Y axis of the heatmap from "frequency" and "alphabetical", with "frequency" as default.
<code>colorPalette</code>	color palette to be used. The default "Spectral" palette shows a color temperature scheme from blue, through yellow, to red (see "Spectral" palette in <code>RColorBrewer</code> 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 <code>Viridis</code> 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 "yymmww" (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

<code>igtProjection</code>	of class <code>IGTProjection</code>
<code>dimensions</code>	number of dimensions of the plot, 2 or 3 (3 by default)
<code>startDate</code>	a Date object indicating the first date to be displayed in the IGT plot. By default the first date of the <code>IGTProjection</code> object.
<code>endDate</code>	a Date object indicating the last date to be displayed in the IGT plot By default the last date of the <code>IGTProjection</code> object.
<code>colorPalette</code>	color palette to be used. The default "Spectral" palette shows a color temperature scheme from blue, through yellow, to red (see "Spectral" palette in <code>RColorBrewer</code> 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 <code>Viridis</code> package).
<code>trajectory</code>	whether to show an estimated trajectory of the information evolution over time. By default FALSE.

Details

Note that since the projection is based on Multi Dimensional 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

DataTemporalMap
 (DataTemporalMap-class), [2](#)
DataTemporalMap, DataTemporalMap-class
 (DataTemporalMap-class), [2](#)
DataTemporalMap-class, [2](#)

estimateDataTemporalMap, [3](#)
estimateIGTProjection, [5](#)
estimateIGTProjection, DataTemporalMap-method
 (estimateIGTProjection), [5](#)
estimateIGTProjection, IGTProjection-method
 (estimateIGTProjection), [5](#)
estimateIGTTrajectory, [7](#)

formatDate, [7](#)

icd9toPheWAS, [8](#)

plotDataTemporalMap, [9](#)
plotDataTemporalMap, DataTemporalMap-method
 (plotDataTemporalMap), [9](#)
plotIGTProjection, [11](#)
plotIGTProjection, IGTProjection-method
 (plotIGTProjection), [11](#)

trimDataTemporalMap, [13](#)
trimDataTemporalMap, DataTemporalMap-method
 (trimDataTemporalMap), [13](#)