

Package ‘ClustAll’

September 18, 2024

Type Package

Title ClustAll: Data driven strategy to find groups of patients within complex diseases

Version 1.1.1

Imports FactoMineR, bigstatsr, clValid, doSNOW, parallel, foreach, dplyr, fpc, mice, modeest, flock, networkD3, methods, ComplexHeatmap, cluster, RColorBrewer, circlize, grDevices, ggplot2, grid, stats, utils, pbapply

Suggests RUnit, knitr, BiocGenerics, rmarkdown, BiocStyle, roxygen2

Description Data driven strategy to find hidden groups of patients with complex diseases using clinical data. ClustAll facilitates the unsupervised identification of multiple robust stratifications. ClustAll, is able to overcome the most common limitations found when dealing with clinical data (missing values, correlated data, mixed data types).

Depends R (>= 4.2.0)

License GPL-2

Encoding UTF-8

biocViews Software, StatisticalMethod, Clustering, DimensionReduction, PrincipalComponent

RoxygenNote 7.3.2

VignetteBuilder knitr

git_url <https://git.bioconductor.org/packages/ClustAll>

git_branch devel

git_last_commit 5d6f959

git_last_commit_date 2024-09-13

Repository Bioconductor 3.20

Date/Publication 2024-09-18

Author Asier Ortega-Legarreta [aut, cre]
(<<https://orcid.org/0009-0000-3563-5362>>),
Sara Palomino-Echeverria [aut]

Maintainer Asier Ortega-Legarreta <aortegal@navarra.es>

Contents

addValidationData	2
characterOrNA	3
ClustAllObject-class	3
cluster2data	4
createClustAll	5
dataImputed	6
dataOriginal	7
dataValidation	7
extractData	8
extractResults	9
initialize,ClustAllObject-method	9
JACCARD_DISTANCE_F	10
listOrNULL	11
logicalOrNA	11
matrixOrNULL	12
nImputation	12
numericOrCharacter	13
numericOrNA	13
obj_noNA1	13
obj_noNA1simplify	14
obj_noNA1 Validation	14
plotJACCARD	15
plotSANKEY	16
processed	17
resStratification	17
runClustAll	18
show,ClustAllObject-method	19
showData	20
summary_clusters	21
validateStratification	21
wdbc	22
wdbcMIDS	23
wdbcNA	24
Index	25

addValidationData	<i>addValidationData</i>
-------------------	--------------------------

Description

Generic function to add validation data to the [ClustAllObject-class](#) object.

Usage

```
addValidationData(Object, dataValidation)
```

Arguments

Object [ClustAllObject-class](#) object.

dataValidation A numeric or character vector with the validation data (true labels). The length of the vector must be the same as input data used in [createClustAll](#).

Value

[ClustAllObject-class](#) object.

See Also

[ClustAllObject-class](#)

Examples

```
data("BreastCancerWisconsin", package = "ClustAll")
label <- as.numeric(as.factor(wdbc$Diagnosis))
wdbc <- wdbc[,-c(1, 2)] # delete patients IDs & label
obj_noNA <- createClustAll(data = wdbc)
obj_noNA <- addValidationData(Object = obj_noNA,
                             dataValidation = label)
```

characterOrNA	<i>characterOrNA Class union of character, null and missing</i>
---------------	---

Description

Contains either character, NULL or missing object

Value

characterOrNA class object

ClustAllObject-class	<i>ClustAllObject</i>
--------------------------------------	-----------------------

Description

This class contains the original data and the imputed datasets needed to run the ClustAll pipeline. The results of all the stratifications are stored in `summary_clusters`. From the stratifications that pass the bootstrapping, a matrix with the Jaccard distances is calculated and stored in `JACCARD_DISTANCE_F`.

Value

[ClustAllObject](#) class object.

Slots

`data` Data Frame of the input data after applying one-hot encoding to the categorical variables and extracting the validation (true label) column.

`dataOriginal` Data Frame of the input data.

`dataImputed` Mids object derived from the mice package that stores the imputed data, in case imputation was applied. Otherwise NULL.

`dataValidation` A vector in the case there is a validation (true label) column in the input data. This information can be added later with [addValidationData](#). Otherwise NULL.

`nImputation` Number of imputations performed.

`processed` A boolean. TRUE if [runClustAll](#) has been executed. Otherwise FALSE.

`summary_clusters` List with the resulting stratifications for each combination of clustering methods (distance + clustering algorithm) and depth, in case [runClustAll](#) has been executed previously. Otherwise NULL.

`JACCARD_DISTANCE_F` Matrix containing the Jaccard distances derived from the robust stratifications after applying the bootstrapping if [runClustAll](#) has been executed previously. Otherwise NULL.

cluster2data

cluster2data: Export selected stratification(s)

Description

Returns the original input data in a Data Frame, appending the selected robust stratification(s) as additional columns. The names of the representative stratifications can be obtained using the [resStratification](#) method.

Usage

```
cluster2data(Object,
              stratificationName)
```

Arguments

`Object` [ClustAllObject-class](#) object.

`stratificationName`

Name of the stratification(s) to be exported.

Value

Returns the original Data Frame with additional column(s) corresponding to the selected stratification(s).

See Also

[resStratification](#), [plotJACCARD](#), [ClustAllObject-class](#)

Examples

```

data("BreastCancerWisconsin", package = "ClustAll")
wdbc <- subset(wdbc,select=c(-ID, -Diagnosis))
wdbc <- wdbc[1:15,1:8]
obj_noNA <- createClustAll(data = wdbc)

obj_noNA1 <- runClustAll(Object = obj_noNA, threads = 1, simplify = TRUE)
resStratification(Object = obj_noNA1, population = 0.05,
                  stratification_similarity = 0.88, all = FALSE)
df <- cluster2data(Object = obj_noNA1,
                  stratificationName = c("cuts_a_1", "cuts_b_5", "cuts_a_5"))

```

createClustAll	<i>createClustAll: Creates ClustAllObject</i>
----------------	---

Description

Creates the `ClustAllObject` object. Applies one-hot encoding to columns with categorical values from the input data and extracts the validation column if available. It performs data imputation if the dataset contains missing values and no imputed dataset is provided. In total there are 3 scenarios when we create the object: - Scenario 1: data does not contain missing values. - Scenario 2: data contains NAs and there is no imputed data. Then, it performs the imputations automatically. - Scenario 3: data contains NAs and imputed data is provided manually. Once `ClustAllObject-class` has been created, the ClustALL pipeline can be run executing `runClustAll`.

Usage

```

createClustAll(data=data,
               nImputation=NULL,
               dataImputed=NULL,
               colValidation=NULL)

```

Arguments

<code>data</code>	Data Frame of the using data. It may contain missing values (NA).
<code>nImputation</code>	Number of imputations to be computed in case the data contains NAs and imputed data is not available.
<code>dataImputed</code>	mids object created with mice package. The input data for the imputation and the data must be the same.
<code>colValidation</code>	vector with the referece labeling of the original dataset provided in "data" (if available). Default is NULL.

Value

`ClustAllObject-class` object.

See Also

`runClustAll`, `ClustAllObject-class`

Examples

```
# Scenario 1: data does not contain missing values
data("BreastCancerWisconsin", package = "ClustAll")
wdbc <- wdbc[,-c(1,2)]
obj_noNA <- createClustAll(data = wdbc)

# Scenario 2: data contains NAs and there is no imputed data.
# Then it performs the imputations automatically
data("BreastCancerWisconsinMISSING", package = "ClustAll")

obj_NA <- createClustAll(wdbcNA, nImputation = 2)

# Scenario 3: data contains NAs and imputed data is provided manually
data("BreastCancerWisconsinMISSING", package = "ClustAll")
ini <- mice::mice(wdbcNA, maxit = 0, print = FALSE)
pred <- ini$pred # predictor matrix
pred["radius1", c("perimeter1", "area1", "smoothness1")] <- 0 # example of
# how to remove predictors

imp <- mice::mice(wdbcNA, m=5, pred=pred, maxit=5, seed=1234, print=FALSE)
obj_imp <- createClustAll(data=wdbcNA, dataImputed = imp)
```

dataImputed

dataImputed accession method

Description

Generic function to retrieve the imputed data obtained in `createClustAll` from a `ClustAllObject-class` object

Usage

```
dataImputed(Object)
```

Arguments

Object `ClustAllObject-class` object

Value

Mids class object with the imputed data or NULL.

See Also

`createClustAll`, `ClustAllObject-class`, `runClustAll`

Examples

```
data("BreastCancerWisconsinMISSING", package = "ClustAll")
data("BreastCancerWisconsin", package = "ClustAll")
wdbc <- subset(wdbc,select=-ID)
obj_NA <- createClustAll(data = wdbcNA, colValidation = "Diagnosis",
                        dataImputed = wdbcMIDS)
dataImputed(obj_NA)
```

dataOriginal

dataOriginal accession method

Description

Generic function to retrieve the initial data used for `createClustAll` from a `ClustAllObject-class` object.

Usage

```
dataOriginal(Object)
```

Arguments

Object `ClustAllObject-class` object.

Value

The Data Frame with the initial data

See Also

[createClustAll](#), [ClustAllObject-class](#), [runClustAll](#)

Examples

```
data("BreastCancerWisconsin", package = "ClustAll")
wdbc <- subset(wdbc,select=-ID)
obj_noNA <- createClustAll(data = wdbc, colValidation = "Diagnosis")
dataOriginal(obj_noNA)
```

dataValidation

dataValidation accession method

Description

Generic function to retrieve numeric vector if it has been added with the true labels from a `ClustAllObject-class` object.

Usage

```
dataValidation(Object)
```

Arguments

Object [ClustAllObject-class](#) object.

Value

numeric vector with true labels if validation column has been added. Otherwise NULL.

See Also

[ClustAllObject-class](#)

Examples

```
data("BreastCancerWisconsin", package = "ClustAll")
wdbc <- subset(wdbc,select=-ID)
obj_noNA <- createClustAll(data = wdbc, colValidation="Diagnosis")
dataValidation(obj_noNA)
```

extractData

extractData

Description

Generic function to retrieve all the data used in [createClustAll](#) from a [ClustAllObject-class](#) object.

Usage

```
extractData(Object)
```

Arguments

Object [ClustAllObject-class](#) object.

Value

List with the information of original, modified and imputed data.

See Also

[createClustAll](#), [ClustAllObject-class](#), [runClustAll](#)

Examples

```
data("BreastCancerWisconsin", package = "ClustAll")
wdbc <- subset(wdbc,select=-ID)

obj_noNA <- createClustAll(data = wdbc, colValidation = "Diagnosis")
extractData(obj_noNA)
```

extractResults	<i>extractResults</i>
----------------	-----------------------

Description

Generic function to retrieve all the results after processing `runClustAll` from a `ClustAllObject-class` object.

Usage

```
extractResults(Object)
```

Arguments

Object `ClustAllObject-class` object.

Value

List with all the names of generated stratifications and the statically robust ones.

See Also

`createClustAll`, `ClustAllObject-class`, `runClustAll`

Examples

```
data("BreastCancerWisconsin", package = "ClustAll")
wdbc <- subset(wdbc, select=-ID)

obj_noNA <- createClustAll(data = wdbc, colValidation = "Diagnosis")
obj_noNA1 <- runClustAll(Object = obj_noNA, threads = 1, simplify = TRUE)
extractResults(obj_noNA1)
```

initialize,ClustAllObject-method
<i>initializeClustAllObject</i>

Description

constructor for `ClustAllObject-class`

Usage

```
## S4 method for signature 'ClustAllObject'
initialize(
  .Object,
  data,
  dataOriginal,
  dataImputed,
  dataValidation,
  nImputation,
  processed,
  summary_clusters,
  JACCARD_DISTANCE_F
)
```

Arguments

<code>.Object</code>	initializing object
<code>data</code>	Data Frame of the input data after applying one-hot encoding to the categorical variables and extracting the validation (true label) column.
<code>dataOriginal</code>	Data Frame of the input data.
<code>dataImputed</code>	Mids object derived from the mice package that stores the imputed data, in case imputation was applied. Otherwise NULL.
<code>dataValidation</code>	A vector in the case there is a validation (true label) column in the input data. This information can be added later with <code>addValidationData</code> . Otherwise NULL.
<code>nImputation</code>	Number of imputations performed.
<code>processed</code>	A boolean. TRUE if <code>runClustAll</code> has been executed. Otherwise FALSE.
<code>summary_clusters</code>	List with the resulting stratifications for each combination of clustering methods (distance + clustering algorithm) and depth, in case <code>runClustAll</code> has been executed previously. Otherwise NULL.
<code>JACCARD_DISTANCE_F</code>	Matrix containing the Jaccard distances derived from the robust stratifications after applying the bootstrapping if <code>runClustAll</code> has been executed previously. Otherwise NULL.

Value

ClustAllObject class object.

JACCARD_DISTANCE_F *JACCARD_DISTANCE_F* accession method

Description

Generic function to retrieve the matrix with the Jaccard distances derived from the robust populations stratifications in `runClustAll` from a `ClustAllObject-class` object.

Usage

```
JACCARD_DISTANCE_F(Object)
```

Arguments

Object [ClustAllObject-class](#) object.

Value

Matrix containing the Jaccard distances derived from the robust populations stratifications or NULL if runClustAll method has not been executed yet.

See Also

[runClustAll](#), [ClustAllObject-class](#)

Examples

```
data("BreastCancerWisconsin", package = "ClustAll")
wdbc <- subset(wdbc,select=c(-ID, -Diagnosis))
wdbc <- wdbc[1:15,1:8]
obj_noNA <- createClustAll(data = wdbc)
obj_noNA1 <- runClustAll(Object = obj_noNA, threads = 1, simplify = FALSE)
JACCARD_DISTANCE_F(obj_noNA1)
```

listOrNULL

Class Union listOrNULL

Description

Contains either list, NULL or missing object

Details

Class union of list, null and missing

Value

listOrNULL class object

logicalOrNA

logicalOrNA

Description

Contains either logical, NULL or missing object

Details

Class union of logical, null and missing

Value

logicalOrNA class object

matrixOrNULL	<i>matrixOrNULL</i>
--------------	---------------------

Description

Contains either matrix or NULL object

Details

Class union of matrix, null and missing

Value

matrixOrNULL class object

nImputation	<i>nImputation</i> accession method
-------------	-------------------------------------

Description

Generic function to retrieve the number of imputations in [createClustAll](#) from a [ClustAllObject-class](#) object.

Usage

```
nImputation(Object)
```

Arguments

Object [ClustAllObject-class](#) object.

Value

Numeric vector that contains the number of imputations. 0 in the case of no imputations were required.

See Also

[createClustAll](#), [ClustAllObject-class](#), [runClustAll](#)

Examples

```
data("BreastCancerWisconsinMISSING", package = "ClustAll")
data("BreastCancerWisconsin", package = "ClustAll")
wdbc <- subset(wdbc,select=-ID)
obj_NA <- createClustAll(data = wdbcNA, colValidation = "Diagnosis",
                        dataImputed = wdbcMIDS)
nImputation(obj_NA)
```

numericOrCharacter	<i>numericOrCharacter</i>
--------------------	---------------------------

Description

Contains either numeric or character object

Details

Class union of numeric and character

Value

numericOrCharacter class object

numericOrNA	<i>Class Union numericOrNA</i>
-------------	--------------------------------

Description

Contains either numeric, NULL or missing object

Details

Class union of numeric, null and missing

Value

numericOrNA class object

obj_noNA1	<i>obj_noNA1: Processed wdbc dataset for testing purposed</i>
-----------	---

Description

Processed wdbc as appear in vignette

Usage

```
data("testData", package = "ClustAll")
```

Format

A processed ClustAllObject

Value

ClustAllObject Object

obj_noNA1simplify *obj_noNA1simplify: Processed wdbc dataset for testing purposed*

Description

Processed wdbc as appear in vignette, with simplify TRUE parameter

Usage

```
data("testData", package = "ClustAll")
```

Format

A processed ClustAllObject

Value

ClustAllObject Object

obj_noNA1Validation *obj_noNA1Validation: Processed wdbc dataset for testing purposed*

Description

Processed wdbc as appear in vignette, with no validation data

Usage

```
data("testData", package = "ClustAll")
```

Format

A processed ClustAllObject

Value

ClustAllObject Object

plotJACCARD	<i>plotJACCARD: Heatmap of robust stratification distances based on Jaccard similarity</i>
-------------	--

Description

This function plots the correlation matrix heatmap showing the Jaccard distances between robust stratifications.

Usage

```
plotJACCARD(Object,  
             paint=TRUE,  
             stratification_similarity=0.7)
```

Arguments

Object	ClustAllObject-class object.
paint	TRUE for painting a square Logical vector with the annotation for the different stratifications.
stratification_similarity	Numeric value representing the minimum Jaccard distance required to consider two stratifications as similar. The default is 0.7.

Value

Plot of a heatmap of the correlation matrix displaying Jaccard distances between statistically robust stratifications.

See Also

[resStratification](#), [cluster2data](#), [ClustAllObject-class](#)

Examples

```
data("BreastCancerWisconsin", package = "ClustAll")  
wdbc <- subset(wdbc,select=c(-ID, -Diagnosis))  
wdbc <- wdbc[1:15,1:8]  
obj_noNA <- createClustAll(data = wdbc)  
  
obj_noNA1 <- runClustAll(Object = obj_noNA, threads = 1, simplify = TRUE)  
plotJACCARD(obj_noNA1, paint = TRUE, stratification_similarity = 0.9)
```

plotSANKEY	<i>plotSANKEY: Plots Sankey Diagram showing the cluster distribution and shifts between a pair of stratifications derived from ClustAllObject</i>
------------	---

Description

This function generates a Sankey Diagram that visualizes the distribution of stratifications and the shifts between two stratifications derived from a ClustAllObject.

Usage

```
plotSANKEY(Object,
            clusters,
            validationData=FALSE)
```

Arguments

Object	ClustAllObject-class object.
clusters	Character vector with the names of a pair of stratifications to compare. When validationData is TRUE, only include the cluster names you want to compare with the validation information. Check resStratification for stratification results.
validationData	Logical value indicating whether to use true labels (validation data) for comparison with the selected stratification.

Value

Sankey plot showing the distribution and shifts between the selected stratifications. When validationData is TRUE, the plot includes the true labels (validation data) if available.

See Also

[resStratification](#), [cluster2data](#), [ClustAllObject-class](#)

Examples

```
data("BreastCancerWisconsin", package = "ClustAll")
label <- as.numeric(as.factor(wdbc$Diagnosis))
wdbc <- subset(wdbc, select=c(-ID, -Diagnosis))
wdbc <- wdbc[1:15, 1:8]
label <- label[16:30]
obj_noNA <- createClustAll(data = wdbc)

obj_noNA1 <- runClustAll(Object = obj_noNA, threads = 1, simplify = TRUE)
resStratification(Object = obj_noNA1, population = 0.05,
                  stratification_similarity = 0.88, all = FALSE)
plotSANKEY(Object = obj_noNA1, clusters = c("cuts_a_1", "cuts_b_5"))

obj_noNA1 <- addValidationData(obj_noNA1, label)
plotSANKEY(Object = obj_noNA1, clusters = "cuts_a_1", validationData=TRUE)
```

processed	<i>processed accession method</i>
-----------	-----------------------------------

Description

Generic function to know if [ClustAllObject-class](#) has been processed. TRUE if [runClustAll](#) has been executed. Otherwise FALSE.

Usage

```
processed(Object)
```

Arguments

Object [ClustAllObject-class](#) object.

Value

TRUE if [runClustAll](#) has been executed. Otherwise FALSE.

See Also

[runClustAll](#), [ClustAllObject-class](#)

Examples

```
data("BreastCancerWisconsin", package = "ClustAll")
wdbc <- subset(wdbc,select=c(-ID, -Diagnosis))
wdbc <- wdbc[1:15,1:8]
obj_noNA <- createClustAll(data = wdbc)
processed(obj_noNA)
```

resStratification	<i>resStratification: Show the representative stratifications</i>
-------------------	---

Description

This function returns a list of representative stratifications. Stratifications that do not meet the minimum population threshold for each group are discarded. The minimum population percentage can be set with a default value of 0.05 (5 robust stratifications; otherwise, it returns the representative for each group of stratification).

Usage

```
resStratification(Object,
                   population=0.05,
                   all=FALSE,
                   stratification_similarity=0.7)
```

Arguments

Object	ClustAllObject-class object.
population	Numeric vector specifying the minimum percentage of the total population that a stratification must have to be considered valid.
all	Logical vector indicating whether to return all representative stratifications for each group of clusters. If FALSE, only the centroid (representative) stratification of each group is returned.
stratification_similarity	Numeric value representing the minimum Jaccard distance required to consider two stratifications as similar. The default is 0.7.

Value

List containing all statistically robust stratifications, grouped by clusters formed based on the previously defined minimum cluster similarity and population criteria. If all is set to TRUE, the function returns all robust stratifications for each cluster. If all is set to FALSE, only the centroid (representative) stratification for each cluster group is returned.

See Also

[plotJACCARD](#), [cluster2data](#), [ClustAllObject-class](#)

Examples

```
data("BreastCancerWisconsin", package = "ClustAll")
wdbc <- subset(wdbc, select=c(-ID, -Diagnosis))
wdbc <- wdbc[1:15, 1:8]
obj_noNA <- createClustAll(data = wdbc)

obj_noNA1 <- runClustAll(Object = obj_noNA, threads = 1, simplify = TRUE)
resStratification(Object = obj_noNA1, population = 0.05,
  stratification_similarity = 0.88, all = FALSE)
```

runClustAll

ClustAll: Data driven strategy to find hidden subgroups of patients within complex diseases using clinical data

Description

This method runs the ClustALL pipeline. The ClustALL framework involves three main steps:

- Step 1: Data Complexity Reduction (DCR): multiple data embeddings are created to replace a highly correlated set of variables with lower-dimension projections derived from Principal Component Analysis (PCA). This process explores all relevant groupings derived from a hierarchical clustering-based dendrogram. Consequently, DCR computes an embedding for each depth in the dendrogram.

- Step 2: The Stratification Process (SP): ClustALL calculates and preliminarily evaluates stratifications for each embedding by computing a stratification for each feasible combination of embedding, dissimilarity metric, and clustering method, considering a predefined range of cluster numbers. The optimal number of clusters is determined using three internal validation measures: the

sum-of-squares (WB-ratio), Dunn index, and average silhouette width. Each combination yields a stratification comprising 'embedding + distance metric + clustering method'.

- Step 3: Consensus-based Stratifications (CbS): filters out non-robust stratifications through bootstrapping. Any stratifications with less than 85 stability are excluded. From the remaining stratifications, representatives of very similar outcomes are selected.

If the simplify parameter is TRUE, every fourth depth of the dendrogram is calculated from the first and second steps. It is helpful in reducing the execution time and obtaining preliminary results. However, it is recommended that it be set to FALSE to obtain robust results.

Usage

```
runClustAll(Object,  
            threads=1,  
            simplify=FALSE)
```

Arguments

Object	ClustAllObject-class object.
threads	numeric vector that indicates the number of cores to use.
simplify	if TRUE, only every fourth depth of the dendrogram will be calculated. It will reduce the execution time, but it is recommended to be set as FALSE to obtain robust results.

Value

A processed object [ClustAllObject-class](#) object.

See Also

[resStratification](#), [plotJACCARD](#), [cluster2data](#), [ClustAllObject-class](#)

Examples

```
data("BreastCancerWisconsin", package = "ClustAll")  
wdbc <- subset(wdbc,select=c(-ID, -Diagnosis))  
wdbc <- wdbc[1:15,1:8]  
  
obj_noNA <- createClustAll(data = wdbc)  
obj_noNA1 <- runClustAll(Object = obj_noNA, threads = 1, simplify = TRUE)  
obj_noNA1
```

show,ClustAllObject-method

show method for ClustAllObject

Description

Show method for a [ClustAllObject-class](#) object

Usage

```
## S4 method for signature 'ClustAllObject'  
show(object)
```

Arguments

object [ClustAllObject-class](#) object

Value

summarize information about the object

showData	<i>data accession method</i>
----------	------------------------------

Description

Generic function to retrieve the initial data used for [createClustAll](#) from a [ClustAllObject-class](#) object.

Usage

```
showData(Object)
```

Arguments

Object [ClustAllObject-class](#) object.

Value

The Data Frame with the initial data

See Also

[createClustAll](#), [ClustAllObject-class](#), [runClustAll](#).

Examples

```
data("BreastCancerWisconsin", package = "ClustAll")  
wdbc <- subset(wdbc,select=-ID)  
obj_noNA <- createClustAll(data = wdbc, colValidation = "Diagnosis")  
showData(obj_noNA)
```

summary_clusters	<i>summary_clusters</i> accession method
------------------	--

Description

Generic function to retrieve the resulting stratifications for each combination of clusterings method (distance + clustering algorithm) and depth of `runClustAll` from a `ClustAllObject-class` object.

Usage

```
summary_clusters(Object)
```

Arguments

Object `ClustAllObject-class` object.

Value

List with the resulting stratifications for each combination of clusterings method (distance + clustering algorithm) and depth methods or NULL if `runClustAll` method has not been executed yet.

See Also

`runClustAll`, `ClustAllObject-class`

Examples

```
data("BreastCancerWisconsin", package = "ClustAll")
wdbc <- subset(wdbc,select=c(-ID, -Diagnosis))
wdbc <- wdbc[1:15,1:8]
obj_noNA <- createClustAll(data = wdbc)
obj_noNA1 <- runClustAll(Object = obj_noNA, threads = 1, simplify = FALSE)
summary_clusters(obj_noNA1)
```

validateStratification

validateStratification: calculates the sensitivity and specificity of selected stratification

Description

Returns the sensitivity and specificity of the selected stratification compared to the true labels (validation data). The names of representative stratifications can be obtained using the method `resStratification`.

Usage

```
validateStratification(Object,
                       stratificationName)
```

Arguments

Object [ClustAllObject-class](#) object.

stratificationName
Character vector with the name of a stratification. Check [resStratification](#) to obtain stratification names.

Value

the sensitivity and specificity values of the selected stratification when validation data is available.

See Also

[resStratification](#), [plotJACCARD](#), [ClustAllObject-class](#)

Examples

```
data("BreastCancerWisconsin", package = "ClustAll")
label <- as.numeric(as.factor(wdbc$Diagnosis))
wdbc <- subset(wdbc, select=c(-ID, -Diagnosis))
wdbc <- wdbc[1:15, 1:8]
label <- label[16:30]
obj_noNA <- createClustAll(data = wdbc)

obj_noNA1 <- runClustAll(Object = obj_noNA, threads = 1, simplify = TRUE)
resStratification(Object = obj_noNA1, population = 0.05,
                  stratification_similarity = 0.88, all = FALSE)
obj_noNA1 <- addValidationData(Object = obj_noNA1,
                              dataValidation = label)
validateStratification(obj_noNA1, "cuts_a_1")
```

wdbc

wdbc: Diagnostic Wisconsin Breast Cancer Database.

Description

A dataset containing Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe the characteristics of the cell nuclei present in the image.

Usage

```
data("BreastCancerWisconsin", package = "ClustAll")
```

Format

A data frame with 660 rows and 31 variables

Details

The dataset comprises two types of features —categorical and numerical— derived from a digitized image of a fine needle aspirate (FNA) of a breast mass from 659 patients. Each patient is characterized by 31 features (10x3) and belongs to one of two target classes: ‘malignant’ or ‘benign’.

Value

wdbc dataset

Source

<<https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic>>

- Diagnosis Label says tumor is malignant or benignant
- radius. Mean of distances from the center to points on the perimeter
- perimeter
- area
- smoothness. Local variation in radius lengths
- compactness. $(\text{Perimeter}^2 / \text{Area}) - 1.0$
- concavity. Severity of concave portions of the contour
- concave points. Number of concave portions of the contour
- symmetry.
- fractal dimension. "Coastline approximation" - 1.

wdbcMIDS

wdbcMIDS: Diagnostic Wisconsin Breast Cancer Database with imputed values

Description

We introduced imputed random values to the wdbcNA dataset. Using Mice. It is a mids object. [wdbc](#)

Usage

```
data("BreastCancerWisconsinMISSING", package = "ClustAll")
```

Format

A data frame with 660 rows and 31 variables

Value

wdbcMIDS dataset

wdbcNA	<i>wdbcNA: Diagnostic Wisconsin Breast Cancer Database with missing values</i>
--------	--

Description

We introduced random missing values to the wdbc dataset. [wdbc](#)

Usage

```
data("BreastCancerWisconsinMISSING", package = "ClustAll")
```

Format

A data frame with 660 rows and 31 variables

Value

wdbcNA dataset

Index

* datasets

obj_noNA1, [13](#)
obj_noNA1simplify, [14](#)
obj_noNAno1Validation, [14](#)
wdbc, [22](#)
wdbcMIDS, [23](#)
wdbcNA, [24](#)

addValidationData, [2](#), [4](#), [10](#)

addValidationData, ClustAllObject, numericOrCharacter-method
(addValidationData), [2](#)

characterOrNA, [3](#)

characterOrNA-class (characterOrNA), [3](#)

ClustAllObject-class, [3](#)

cluster2data, [4](#), [15](#), [16](#), [18](#), [19](#)

cluster2data, ClustAllObject, character-method
(cluster2data), [4](#)

createClustAll, [3](#), [5](#), [6-9](#), [12](#), [20](#)

createClustAll, data.frame, numericOrNA, ANY, characterOrNA-method
(createClustAll), [5](#)

dataImputed, [6](#)

dataImputed, ClustAllObject-method
(dataImputed), [6](#)

dataOriginal, [7](#)

dataOriginal, ClustAllObject-method
(dataOriginal), [7](#)

dataValidation, [7](#)

dataValidation, ClustAllObject-method
(dataValidation), [7](#)

extractData, [8](#)

extractData, ClustAllObject-method
(extractData), [8](#)

extractResults, [9](#)

extractResults, ClustAllObject-method
(extractResults), [9](#)

initialize, ClustAllObject-method, [9](#)

JACCARD_DISTANCE_F, [10](#)

JACCARD_DISTANCE_F, ClustAllObject-method
(JACCARD_DISTANCE_F), [10](#)

listOrNULL, [11](#)

listOrNULL-class (listOrNULL), [11](#)

logicalOrNA, [11](#)

logicalOrNA-class (logicalOrNA), [11](#)

matrixOrNULL, [12](#)

matrixOrNULL-class (matrixOrNULL), [12](#)

nImputation, [12](#)

nImputation, ClustAllObject-method
(nImputation), [12](#)

numericOrCharacter, [13](#)

numericOrCharacter-class
(numericOrCharacter), [13](#)

numericOrNA, [13](#)

numericOrNA-class (numericOrNA), [13](#)

obj_noNA1, [13](#)

obj_noNA1simplify, [14](#)

obj_noNAno1Validation, [14](#)
obj_noNAno1Validation, ClustAllObject-method

plotJACCARD, [4](#), [15](#), [18](#), [19](#), [22](#)

plotJACCARD, ClustAllObject, logicalOrNA, numericOrNA-method
(plotJACCARD), [15](#)

plotSANKEY, [16](#)

plotSANKEY, ClustAllObject, character, logicalOrNA-method
(plotSANKEY), [16](#)

processed, [17](#)

processed, ClustAllObject-method
(processed), [17](#)

resStratification, [4](#), [15](#), [16](#), [17](#), [19](#), [21](#), [22](#)

resStratification, ClustAllObject, numericOrNA, logicalOrNA-method
(resStratification), [17](#)

runClustAll, [4-12](#), [17](#), [18](#), [20](#), [21](#)

runClustAll, ClustAllObject, numericOrNA, logicalOrNA-method
(runClustAll), [18](#)

show, ClustAllObject-method, [19](#)

showData, [20](#)

showData, ClustAllObject-method
(showData), [20](#)

summary_clusters, [21](#)

summary_clusters, ClustAllObject-method
(summary_clusters), [21](#)

validateStratification, [21](#)
validateStratification, ClustAllObject, characterOrNA-method
 (validateStratification), [21](#)

wdbc, [22](#), [23](#), [24](#)
wdbcMIDS, [23](#)
wdbcNA, [24](#)