Package 'coca'

October 12, 2022

Title Cluster-of-Clusters Analysis

Version 1.1.0

Author Alessandra Cabassi [aut, cre] (<https://orcid.org/0000-0003-1605-652X>), Paul DW Kirk [ths] (<https://orcid.org/0000-0002-5931-7489>)

Description Contains the R functions needed to perform Cluster-Of-Clusters Analysis (COCA) and Consensus Clustering (CC). For further details please see Cabassi and Kirk (2020) <doi:10.1093/bioinformatics/btaa593>.

Depends R (>= 3.5.0)

License MIT + file LICENSE

URL http://github.com/acabassi/coca

BugReports http://github.com/acabassi/coca/issues

Encoding UTF-8

LazyData true

Imports caret, cluster, fpc, glmnet, Matrix, nnet, pheatmap, RColorBrewer, sparcl, stats

Suggests knitr, mclust, rmarkdown

RoxygenNote 7.1.0

VignetteBuilder knitr

NeedsCompilation no

Maintainer Alessandra Cabassi <alessandra.cabassi@mrc-bsu.cam.ac.uk>

Repository CRAN

Date/Publication 2020-07-06 17:00:09 UTC

R topics documented:

`buildMOC` *Build Matrix-Of-Clusters*

Description

This function creates a matrix of clusters starting from a list of heterogeneous datasets.

Usage

```
buildMOC(
  data,
  M,
  K = NULL,
  maxK = 10,
  methods = "hclust",
  distances = "euclidean",
  fill = FALSE,
  computeAccuracy = FALSE,
  fullData = FALSE,
  savePNG = FALSE,
  fileName = "buildMOC",
  widestGap = FALSE,
  dunns = FALSE,
  dunn2s = FALSE
)
```

Arguments

`data`	List of M datasets, each of size N X P_m, where m = 1, ..., M.
`M`	Number of datasets.
`K`	Vector containing the number of clusters in each dataset. If given an integer instead of a vector it is assumed that each dataset has the same number of clusters. If NULL, it is assumed that the true cluster numbers are not known, therefore they will be estimated using the silhouette method.
`maxK`	Vector of maximum cluster numbers to be considered for each dataset if K is NULL. If given an integer instead of a vector it is assumed that for each dataset the same maximum number of clusters must be considered. Default is 10.

`methods` Vector of strings containing the names of the clustering methods to be used to cluster the observations in each dataset. Each can be "kmeans" (k-means clustering), "hclust" (hierarchical clustering), or "pam" (partitioning around medoids). If the vector is of length one, the same clustering method is applied to all the datasets. Default is "hclust".

`distances` Distances to be used in the clustering step for each dataset. If only one string is provided, then the same distance is used for all datasets. If the number of strings provided is the same as the number of datasets, then each distance will be used for the corresponding dataset. Default is "euclidean". Please note that not all distances are compatible with all clustering methods. "euclidean" and "manhattan" work with all available clustering algorithms. "gower" distance is only available for partitioning around medoids. In addition, "maximum", "canberra", "binary" or "minkowski" are available for k-means and hierarchical clustering.

`fill` Boolean. If TRUE, if there are any missing observations in one or more datasets, the corresponding cluster labels will be estimated through generalised linear models on the basis of the available labels.

`computeAccuracy` Boolean. If TRUE, for each missing element, the performance of the predictive model used to estimate the corresponding missing label is computer.

`fullData` Boolean. If TRUE, the full data matrices are used to estimate the missing cluster labels (instead of just using the cluster labels of the corresponding datasets).

`savePNG` Boolean. If TRUE, plots of the silhouette for each datasets are saved as png files. Default is FALSE.

`fileName` If `savePNG` is TRUE, this is the string containing the name of the output files. Can be used to specify the folder path too. Default is "buildMOC". The ".png" extension is automatically added to this string.

`widestGap` Boolean. If TRUE, compute also widest gap index to choose best number of clusters. Default is FALSE.

`dunns` Boolean. If TRUE, compute also Dunn's index to choose best number of clusters. Default is FALSE.

`dunn2s` Boolean. If TRUE, compute also alternative Dunn's index to choose best number of clusters. Default is FALSE.

Value

This function returns a list containing:

`moc` the Matrix-Of-Clusters, a binary matrix of size N x sum(K) where element (n,k) contains a 1 if observation n belongs to the corresponding cluster, 0 otherwise.

`datasetIndicator` a vector of length sum(K) in which each element is the number of the dataset to which the cluster belongs.

`number_nas` the total number of NAs in the matrix of clusters. (If the MOC has been filled with imputed values, `number_nas` indicates the number of NAs in the original MOC.)

clLabels — a matrix that is equivalent to the matrix of clusters, but is in compact form, i.e. each column corresponds to a dataset, each row represents an observation, and its values indicate the cluster labels.

K — vector of cluster numbers in each dataset. If these are provided as input, this is the same as the input (expanded to a vector if the input is an integer). If the cluster numbers are not provided as input, this vector contains the cluster numbers chosen via silhouette for each dataset.

Author(s)

Alessandra Cabassi <alessandra.cabassi@mrc-bsu.cam.ac.uk>

References

The Cancer Genome Atlas, 2012. Comprehensive molecular portraits of human breast tumours. Nature, 487(7407), pp.61–70.

Rousseeuw, P.J., 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics, 20, pp.53-65.

Examples

```
# Load data
data <- list()
data[[1]] <- as.matrix(read.csv(system.file("extdata", "dataset1.csv",
package = "coca"), row.names = 1))
data[[2]] <- as.matrix(read.csv(system.file("extdata", "dataset2.csv",
package = "coca"), row.names = 1))
data[[3]] <- as.matrix(read.csv(system.file("extdata", "dataset3.csv",
package = "coca"), row.names = 1))

# Build matrix of clusters
outputBuildMOC <- buildMOC(data, M = 3, K = 6, distances = "cor")

# Extract matrix of clusters
matrixOfClusters <- outputBuildMOC$moc
```

---

chooseKusingAUC *Choose number of clusters based on AUC*

Description

This function allows to choose the number of clusters in a dataset based on the area under the curve of the empirical distribution function of a consensus matrix, calculated for different (consecutive) cluster numbers, as explained in the article by Monti et al. (2003), Section 3.3.1.

Usage

```
chooseKusingAUC(areaUnderTheCurve, savePNG = FALSE, fileName = "deltaAUC.png")
```

Arguments

areaUnderTheCurve
: Vector of length maxK-1 containing the area under the curve of the empirical distribution function of the consensus matrices obtained with K varying from 2 to maxK.

savePNG
: Boolean. If TRUE, a plot of the area under the curve for each value of K is saved as a png file. The file is saved in a subdirectory of the working directory, called "delta-auc". Default is FALSE.

fileName
: If savePNG is TRUE, this is the name of the png file. Can be used to specify the folder path too. Default is "deltaAUC". The ".png" extension is automatically added to this string.

Value

This function returns a list containing:

deltaAUC
: a vector of length maxK-1 where element i is the area under the curve for K = i+1 minus the area under the curve for K = i (for i = 2 this is simply the area under the curve for K = i)

K
: the lowest among the values of K that are chosen by the algorithm.

Author(s)

Alessandra Cabassi <alessandra.cabassi@mrc-bsu.cam.ac.uk>

References

Monti, S., Tamayo, P., Mesirov, J. and Golub, T., 2003. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. Machine learning, 52(1-2), pp.91-118.

Examples

```
# Assuming that we want to choose among any value of K (number of clusters)
# between 2 and 10 and that the area under the curve is as follows:
areaUnderTheCurve <- c(0.05, 0.15, 0.4, 0.5, 0.55, 0.56, 0.57, 0.58, 0.59)

# The optimal value of K can be chosen with:
K <- chooseKusingAUC(areaUnderTheCurve)$K
```

---

coca *Cluster-Of-Clusters Analysis*

Description

This function allows to do Cluster-Of-Clusters Analysis on a binary matrix where each column is a clustering of the data, each row corresponds to a data point and the element in position (i,j) is equal to 1 if data point i belongs to cluster j, 0 otherwise.

Usage

```
coca(
  moc,
  K = NULL,
  maxK = 6,
  B = 1000,
  pItem = 0.8,
  hclustMethod = "average",
  choiceKmethod = "silhouette",
  ccClMethod = "kmeans",
  ccDistHC = "euclidean",
  maxIterKM = 1000,
  savePNG = FALSE,
  fileName = "coca",
  verbose = FALSE,
  widestGap = FALSE,
  dunns = FALSE,
  dunn2s = FALSE,
  returnAllMatrices = FALSE
)
```

Arguments

moc	N X C data matrix, where C is the total number of clusters considered.
K	Number of clusters.
maxK	Maximum number of clusters considered for the final clustering if K is not known. Default is 6.
B	Number of iterations of the Consensus Clustering step.
pItem	Proportion of items sampled at each iteration of the Consensus Cluster step.
hclustMethod	Agglomeration method to be used by the hclust function to perform hierarchical clustering on the consensus matrix. Can be "single", "complete", "average", etc. For more details please see ?stats::hclust.
choiceKmethod	Method used to choose the number of clusters if K is NULL, can be either "AUC" (area under the curve, work in progress) or "silhouette". Default is "silhouette".
ccClMethod	Clustering method to be used by the Consensus Clustering algorithm (CC). Can be either "kmeans" for k-means clustering or "hclust" for hiearchical clustering. Default is "kmeans".
ccDistHC	Distance to be used by the hiearchical clustering algorithm inside CC. Can be "pearson" (for 1 - Pearson correlation), "spearman" (for 1- Spearman correlation), or any of the distances provided in stats::dist() (i.e. "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowski"). Default is "euclidean".
maxIterKM	Number of iterations for the k-means clustering algorithm. Default is 1000.
savePNG	Boolean. Save plots as PNG files. Default is FALSE.

`fileName` If `savePNG` is TRUE, this is the string containing (the first part of) the name of the output files. Can be used to specify the folder path too. Default is "coca". The ".png" extension is automatically added to this string.

`verbose` Boolean.

`widestGap` Boolean. If TRUE, compute also widest gap index to choose best number of clusters. Default is FALSE.

`dunns` Boolean. If TRUE, compute also Dunn's index to choose best number of clusters. Default is FALSE.

`dunn2s` Boolean. If TRUE, compute also alternative Dunn's index to choose best number of clusters. Default is FALSE.

`returnAllMatrices` Boolean. If TRUE, return consensus matrices for all considered values of K. Default is FALSE.

Value

This function returns a list containing:

`consensusMatrix` a symmetric matrix where the element in position (i,j) corresponds to the proportion of times that items i and j have been clustered together and a vector of cluster labels.

`clusterLabels` the final cluster labels.

`K` the final number of clusters. If provided by the user, this is the same as the input. Otherwise, this is the number of clusters selected via the requested method (see argument `choiceKmethod`).

`consensusMatrices` if returnAllMatrices = TRUE, this array also returned, containing the consensus matrices obtained for each of the numbers of clusters considered by the algorithm.

Author(s)

Alessandra Cabassi <alessandra.cabassi@mrc-bsu.cam.ac.uk>

References

The Cancer Genome Atlas, 2012. Comprehensive molecular portraits of human breast tumours. Nature, 487(7407), pp.61–70.

Cabassi, A. and Kirk, P. D. W. (2019). Multiple kernel learning for integrative consensus clustering of 'omic datasets. arXiv preprint. arXiv:1904.07701.

Examples

```
# Load data
data <- list()
data[[1]] <- as.matrix(read.csv(system.file("extdata", "dataset1.csv",
package = "coca"), row.names = 1))
```

```
data[[2]] <- as.matrix(read.csv(system.file("extdata", "dataset2.csv",
package = "coca"), row.names = 1))
data[[3]] <- as.matrix(read.csv(system.file("extdata", "dataset3.csv",
package = "coca"), row.names = 1))

# Build matrix of clusters
outputBuildMOC <- buildMOC(data, M = 3, K = 5, distances = "cor")

# Extract matrix of clusters
moc <- outputBuildMOC$moc

# Do Cluster-Of-Clusters Analysis
outputCOCA <- coca(moc, K = 5)

# Extract cluster labels
clusterLabels <- outputCOCA$clusterLabels
```

`consensusCluster` *Consensus clustering*

Description

This function allows to perform consensus clustering using the k-means clustering algorithm, for a fixed number of clusters. We consider the number of clusters K to be fixed.

Usage

```
consensusCluster(
  data = NULL,
  K = 2,
  B = 100,
  pItem = 0.8,
  clMethod = "hclust",
  dist = "euclidean",
  hclustMethod = "average",
  sparseKmeansPenalty = NULL,
  maxIterKM = 1000
)
```

Arguments

`data`	N X P data matrix
`K`	Number of clusters.
`B`	Number of iterations.
`pItem`	Proportion of items sampled at each iteration.

`clMethod` Clustering algorithm. Can be "hclust" for hierarchical clustering, "kmeans" for k-means clustering, "pam" for partitioning around medoids, "sparse-kmeans" for sparse k-means clustering or "sparse-hclust" for sparse hierarchical clustering. Default is "hclust". However, if the data contain at least one covariate that is a factor, the default clustering algorithm is "pam".

`dist` Distance used for hierarchical clustering. Can be "pearson" (for 1 - Pearson correlation), "spearman" (for 1- Spearman correlation), any of the distances provided in stats::dist() (i.e. "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowski"), or a matrix containing the distances between the observations.

`hclustMethod` Hierarchical clustering method. Default is "average". For more details see `?hclust`.

`sparseKmeansPenalty`
If the selected clustering method is "sparse-kmeans", this is the value of the parameter "wbounds" of the "KMeansSparseCluster" function. The default value is the square root of the number of variables.

`maxIterKM` Number of iterations for the k-means clustering algorithm.

Value

The output is a consensus matrix, that is a symmetric matrix where the element in position (i,j) corresponds to the proportion of times that items i and j have been clustered together.

Author(s)

Alessandra Cabassi <alessandra.cabassi@mrc-bsu.cam.ac.uk>

References

Monti, S., Tamayo, P., Mesirov, J. and Golub, T., 2003. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. Machine learning, 52(1-2), pp.91-118.

Witten, D.M. and Tibshirani, R., 2010. A framework for feature selection in clustering. Journal of the American Statistical Association, 105(490), pp.713-726.

Examples

```
# Load one dataset with 300 observations, 2 variables, 6 clusters
data <- as.matrix(read.csv(system.file("extdata", "dataset1.csv",
package = "coca"), row.names = 1))

# Compute consensus clustering with K=5 clusters
cm <- consensusCluster(data, K = 5)
```

expandMOC *Expand matrix of clusters*

Description

Expand matrix of cluster labels into matrix of clusters

Usage

```
expandMOC(clLabels, datasetNames = NULL)
```

Arguments

clLabels	Matrix of cluster labels of size N x M.
datasetNames	Vector of cluster names of length M. Default is NULL.

Value

The output is a list containing:

moc	the matrix of clusters.
datasetIndicator	a vector containing the dataset indicator.
datasetNames	an expanded vector of dataset names for the moc.

Author(s)

Alessandra Cabassi <alessandra.cabassi@mrc-bsu.cam.ac.uk>

Examples

```
# Load data
data <- list()
data[[1]] <- as.matrix(read.csv(system.file("extdata", "dataset1.csv",
package = "coca"), row.names = 1))
data[[2]] <- as.matrix(read.csv(system.file("extdata", "dataset2.csv",
package = "coca"), row.names = 1))
data[[3]] <- as.matrix(read.csv(system.file("extdata", "dataset3.csv",
package = "coca"), row.names = 1))

# Build matrix of clusters
outputBuildMOC <- buildMOC(data, M = 3, K = 6, distances = "cor")

# Extract matrix of clusters
clLabels <- outputBuildMOC$clLabels

# Impute missing values
outputFillMOC <- fillMOC(clLabels, data = data)
```

```
# Replace matrix of cluster labels with new (full) one
clLabels <- outputFillMOC$fullClLabels

# Expand matrix of cluster labels into matrix of clusters
outputExpandMOC <- expandMOC(clLabels)
clLabels <- outputExpandMOC$clLabels
```

`fillMOC` *Fill Matrix-Of-Clusters*

Description

This function fills in a matrix of clusters that contains NAs, by estimating the missing cluster labels based on the available ones or based on the other datasets. The predictive accuracy of this method can also be estimated via cross-validation.

Usage

```
fillMOC(clLabels, data, computeAccuracy = FALSE, verbose = FALSE)
```

Arguments

`clLabels`	N X M matrix containing cluster labels. Element (n,m) contains the cluster label for element data point n in cluster m.
`data`	List of M datasets to be used for the label imputation.
`computeAccuracy`	Boolean. If TRUE, for each missing element, the performance of the predictive model used to estimate the corresponding missing label is computer. Default is FALSE.
`verbose`	Boolean. If TRUE, for each NA, the size of the matrix used to estimate its values is printed to screen. Default is FALSE.

Value

The output is a list containing:

`fullClLabels`	the same matrix of clusters as the input matrix `clLabels`, where NAs have been replaced by their estimates, where possible.
`nRows`	matrix where the item in position (i,j) indicates the number of observations used in the predictive model used to estimate the corresponding missing label in the `fullClLabels` matrix.
`nColumns`	matrix where the item in position (i,j) indicates the number of covariates used in the predictive model used to estimate the corresponding missing label in the `fullClLabels` matrix.
`accuracy`	a matrix where each element corresponds to the predictive accuracy of the predictive model used to estimate the corresponding label in the cluster label matrix. This is only returned if the argument `computeAccuracy` is set to TRUE.

accuracy_random

This is computed in the same way as accuracy, but with the labels randomly shuffled. This can be used in order to assess the predictive accuracy of the imputation algorithm and is returned only if the argument computeAccuracy is set to TRUE.

Author(s)

Alessandra Cabassi <alessandra.cabassi@mrc-bsu.cam.ac.uk>

References

The Cancer Genome Atlas, 2012. Comprehensive molecular portraits of human breast tumours. Nature, 487(7407), pp.61–70.

Examples

```
# Load data
data <- list()
data[[1]] <- as.matrix(read.csv(system.file("extdata", "dataset1.csv",
                        package = "coca"), row.names = 1))
data[[2]] <- as.matrix(read.csv(system.file("extdata", "dataset2.csv",
                        package = "coca"), row.names = 1))
data[[3]] <- as.matrix(read.csv(system.file("extdata", "dataset3.csv",
                        package = "coca"), row.names = 1))

# Build matrix of clusters
outputBuildMOC <- buildMOC(data, M = 3, K = 6, distances = "cor")

# Extract matrix of clusters
clLabels <- outputBuildMOC$clLabels

# Impute missing values using full datasets
outputFillMOC <- fillMOC(clLabels, data)

# Extract full matrix of cluster labels
clLabels2 <- outputFillMOC$fullClLabels
```

---

maximiseSilhouette *Choose K that maximises the silhouette from a set of kernel matrices and clusterings*

Description

Choose the number of clusters K that maximises the silhouette, starting from a set of kernel matrices each corresponding to a different choice of K and the corresponding clusterings of the data for each of those values of K.

Usage

```
maximiseSilhouette(
  kernelMatrix,
  clLabels,
  maxK,
  savePNG = FALSE,
  fileName = "silhouette",
  isDistance = FALSE,
  widestGap = FALSE,
  dunns = FALSE,
  dunn2s = FALSE
)
```

Arguments

kernelMatrix	N X N X (maxK-1) array of kernel matrices.
clLabels	(maxK-1) X N matrix containing the clusterings obtained for different values of K.
maxK	Maximum number of clusters considered.
savePNG	If TRUE, a plot of the silhouette is saved in the working folder. Defaults to FALSE.
fileName	If savePNG is TRUE, this is the name of the png file.
isDistance	Boolean. If TRUE, the kernel matrices are interpreted as matrices of distances, otherwise as matrices of similarities.
widestGap	Boolean. If TRUE, also computes widest gap index (and plots it if savePNG is TRUE).
dunns	Boolean. If TRUE, also computes Dunn's index: minimum separation / maximum diameter (and plots it if savePNG is TRUE).
dunn2s	Boolean. If TRUE, also computes an alternative version of Dunn's index: minimum average dissimilarity between two cluster / maximum average within cluster dissimilarity (and plots it if savePNG is TRUE).

Value

The function returns a list containing:

silh	a vector of length maxK-1 such that silh[i] is the silhouette for K = i+1
K	the lowest number of clusters for which the silhouette is maximised.

Author(s)

Alessandra Cabassi <alessandra.cabassi@mrc-bsu.cam.ac.uk>

plotMOC *Plot Matrix-Of-Clusters*

Description

This function creates a matrix of clusters, starting from a list of heterogeneous datasets.

Usage

```
plotMOC(
  moc,
  datasetIndicator,
  datasetNames = NULL,
  annotations = NULL,
  clr = FALSE,
  clc = FALSE,
  savePNG = FALSE,
  fileName = "moc.png",
  showObsNames = FALSE,
  showClusterNames = FALSE,
  annotation_colors = NA
)
```

Arguments

moc	Matrix-Of-Clusters of size N x sumK.
datasetIndicator	Vector containing integers indicating which rows correspond to some clustering of the same dataset.
datasetNames	Vector containing the names of the datasets to which each column of labels corresponds. If NULL, datasetNames will be the same as datasetIndicator. Default is NULL.
annotations	Dataframe containing annotations. Number of rows must be N. If the annotations are integers, use `as.factor()` for a better visual result.
clr	Cluster rows. Default is FALSE.
clc	Cluster columns. Default is FALSE.
savePNG	Boolean. If TRUE, plot is saved as a png file.
fileName	If savePNG is TRUE, this is the string containing the name of the moc figure. Can be used to specify the folder path too. Default is "moc". The ".png" extension is automatically added to this string.
showObsNames	Boolean. If TRUE, the plot will also include the column names (i.e. name of each observation). Default is FALSE, since there are usually too many columns.
showClusterNames	Boolean. If TRUE, plot cluster names next to corresponding row. Default is FALSE.
annotation_colors	Optional. See annotation_colors in pheatmap::pheatmap.

Author(s)

Alessandra Cabassi <alessandra.cabassi@mrc-bsu.cam.ac.uk>

References

The Cancer Genome Atlas, 2012. Comprehensive molecular portraits of human breast tumours. Nature, 487(7407), pp.61–70.

Examples

```
# Load data
data <- list()
data[[1]] <- as.matrix(read.csv(system.file("extdata", "dataset1.csv",
package = "coca"), row.names = 1))
data[[2]] <- as.matrix(read.csv(system.file("extdata", "dataset2.csv",
package = "coca"), row.names = 1))
data[[3]] <- as.matrix(read.csv(system.file("extdata", "dataset3.csv",
package = "coca"), row.names = 1))

# Create vector of dataset names, in the same order as they appear above
datasetNames <- c("Dataset1", "Dataset2", "Dataset3")

# Build matrix of clusters
outputBuildMOC <- buildMOC(data, M = 3, K = 6, distances = "cor")

# Extract matrix of clusters and dataset indicator vector
moc <- outputBuildMOC$moc
datasetIndicator <- outputBuildMOC$datasetIndicator

# Prepare annotations
true_labels <- as.matrix(read.csv(
system.file("extdata", "cluster_labels.csv", package = "coca"),
row.names = 1))
annotations <- data.frame(true_labels = as.factor(true_labels))

# Plot matrix of clusters
plotMOC(moc,
        datasetIndicator,
        datasetNames = datasetNames,
        annotations = annotations)
```

Index