Package ‘SOMEnv’

October 12, 2022

Type Package

Title SOM Algorithm for the Analysis of Multivariate Environmental Data

Version 1.1.2

Maintainer Sabina Licen <slicen@units.it>

Description Analysis of multivariate environmental high frequency data by Self-Organizing Map and k-means clustering algorithms. By means of the graphical user interface it provides a comfortable way to elaborate by self-organizing map algorithm rather big datasets (txt files up to 100 MB ) obtained by environmental high-frequency monitoring by sensors/instruments. The functions present in the package are based on 'kohonen' and 'openair' packages implemented by functions embedding Vesanto et al. (2001) <http://www.cis.hut.fi/projects/somtoolbox/package/papers/techrep.pdf> heuristic rules for map initialization parameters, k-means clustering algorithm and map features visualization. Cluster profiles visualization as well as graphs dedicated to the visualization of time-dependent variables Licen et al. (2020) <doi:10.4209/aaqr.2019.08.0414> are provided.

License GPL-3

Encoding UTF-8

Depends R (>= 3.6.0)

RoxygenNote 7.1.1

Imports rlist, kohonen, shiny, dplyr, plyr, openair, colourpicker, shinycssloaders, shinycustomloader

URL https://github.com/SomEnv/somenv

BugReports https://github.com/SomEnv/somenv/issues

Author Sabina Licen [aut, cre],
Marco Franzon [aut],
Tommaso Rodani [aut],
Pierluigi Barbieri [aut]

NeedsCompilation no

Repository CRAN

Date/Publication 2021-07-26 13:30:02 UTC
**BmusCentr**

**Description**

The function finds the Best Matching Units of the cluster centroids

**Usage**

```r
BmusCentr(centroids, som_model, k)
```
**BmusClus**

**Arguments**

- **centroids**: Centroids array (output of kmeans_clustersR function)
- **som_model**: An object of class kohonen
- **k**: Number of clusters

**Value**

An array containing the BMU for each centroid

**Author(s)**

Sabina Licen

**References**


---

**BmusClus**  
*Cluster assignment for the experimental data*

**Description**

Generate a vector containing the cluster assignment to experimental data

**Usage**

`BmusClus(Bmus, Cluster)`

**Arguments**

- **Bmus**: Best Matching Unit assignment to the experimental data
- **Cluster**: Vector containing cluster number assignment for prototypes

**Value**

A vector containing the cluster assignment to experimental data

**Author(s)**

Sabina Licen

**References**

BoxClus

Boxplot of prototype variables split by cluster and variable

Description

Boxplot function is used, box whiskers are omitted

Usage

BoxClus(Dms, codebook, Cluster, Centroids)

Arguments

Dms
A vector of length 2, where the first argument specifies the number of rows and
the second the number of columns of plots (see mfrow in par)

codebook
De-normalized prototype codebook

Cluster
Vector containing cluster number assignment for prototypes

Centroids
Centroids matrix

Value

Boxplot of prototype variables split by cluster

Author(s)

Sabina Licen

References

10.4209/aaqr.2019.08.0414

See Also

boxplot, par
BoxUnits

Boxplot of prototype variables split by cluster

Description

Boxplot function is used, box whiskers are omitted

Usage

BoxUnits(codebook, Cluster, Centroids, Ylim = NA, pitch = NA, xdim = 0.75)

Arguments

codebook Prototype codebook normalized by variable
Cluster Vector containing cluster number assignment for prototypes
Centroids Centroids matrix
Ylim Vector of length 2 for y-axis limits
pitch Vector containing the position of horizontal grid lines
xdim x axes label dimensions

Value

Boxplot of prototype variables split by cluster

Author(s)

Sabina Licen

References


See Also

boxplot
ClusCol

*Custom color sequence for clusters*

**Description**

Generate the sequence of colors to plot the SOM map according to clusters

**Usage**

```r
ClusCol(Centroids, Cluster, colSeq = rainbow(nrow(data.frame(Centroids))))
```

**Arguments**

- **Centroids**
  Centroids matrix

- **Cluster**
  Vector containing cluster number assignment for prototypes

- **colSeq**
  Color sequence for the clusters

**Value**

A vector of colors with length equal to Cluster

**Author(s)**

Sabina Licen

**References**


---

CodeCoord

*Prototype coordinates for graph*

**Description**

Generate X and Y coordinates for plotting a SOM map shaped according to Vesanto visualization fashion

**Usage**

```r
CodeCoord(Row, Col)
```

**Arguments**

- **Row**
  Number of SOM map rows

- **Col**
  Number of SOM map columns
Value

This function returns a data.frame including columns:

- X
- Y

Author(s)

Sabina Licen, Pierluigi Barbieri

References


Examples

Coord<-CodeCoord(10,5)

---

DailyBar

Plot of daily percentages for each cluster

Description

The function produces a plot representing the the daily percentage for each cluster

Usage

DailyBar(
  experimental,
  TrainClus,
  Centroids,
  colSeq = rainbow(nrow(data.frame(Centroids))),
  Total = 1440,
  xdim = 0.7,
  ydim = 0.8
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>experimental</td>
<td>Experimental data (must contain variable &quot;date&quot;)</td>
</tr>
<tr>
<td>TrainClus</td>
<td>Vector containing cluster number assignment for experimental data</td>
</tr>
<tr>
<td>Centroids</td>
<td>Centroids matrix</td>
</tr>
<tr>
<td>colSeq</td>
<td>Color sequence for the clusters</td>
</tr>
<tr>
<td>Total</td>
<td>Number of observations per day</td>
</tr>
<tr>
<td>xdim</td>
<td>x axes label dimensions</td>
</tr>
<tr>
<td>ydim</td>
<td>y axes label dimensions</td>
</tr>
</tbody>
</table>
Value
Plot of daily percentages for each cluster, the latter element in the legend represents percentage of not determined data

Author(s)
Sabina Licen

References

---

\texttt{db\_indexR} \hspace{2cm} \textit{Evaluate Davis-Bouldin index for the cluster split of data input}

Description
The function has been coded in R code starting from \texttt{db\_index.m} script present in somtoolbox for Matlab by Vesanto and adapted for the use in the shiny app

Usage
\begin{verbatim}
db_indexR(codebook, k_best, c_best)
\end{verbatim}

Arguments
\begin{itemize}
  \item \texttt{codebook} \quad SOM codebook
  \item \texttt{k\_best} \quad Vector with cluster number assignment for each sample
  \item \texttt{c\_best} \quad Matrix with cluster centroids
\end{itemize}

Value
The mean DB-index for the clustering

Author(s)
Sabina Licen, Pierluigi Barbieri

References

See Also
\texttt{som\_mdistR}, \texttt{kmeans\_clustersRProg}
**Freq**

*Percentage frequency for each cluster*

**Description**

Percentage frequency for each cluster

**Usage**

Freq(Cluster, Centroids)

**Arguments**

- **Cluster**: Vector containing cluster number assignment for experimental data
- **Centroids**: Centroids matrix

**Value**

A data frame containing the percentage frequency of each cluster

**Author(s)**

Sabina Licen

**References**


**FreqD**

*Daily percentage frequency for each cluster*

**Description**

Daily percentage frequency for each cluster

**Usage**

FreqD(Date, Cluster, Centroids, Total = 1440)

**Arguments**

- **Date**: Vector containing date/time variable for experimental data
- **Cluster**: Vector containing cluster number assignment for experimental data
- **Centroids**: Centroids matrix
- **Total**: Number of observations per day
**Value**

A data frame containing the daily percentage frequency of each cluster

**Author(s)**

Sabina Licen

**References**


---

**FreqM**

*Monthly percentage frequency for each cluster*

**Description**

Monthly percentage frequency for each cluster

**Usage**

FreqM(Date, Cluster, Centroids)

**Arguments**

- **Date**: Vector containing date/time variable for experimental data
- **Cluster**: Vector containing cluster number assignment for experimental data
- **Centroids**: Centroids matrix

**Value**

A data frame containing the monthly percentage frequency of each cluster

**Author(s)**

Sabina Licen

**References**

Hexa

*Function to draw an hexagon around a point*

**Description**

Draws an hexagon around a point of x and y coordinates

**Usage**

`Hexa(x, y, color = NA, border = "gray", unitcell = 1)`

**Arguments**

- `x`: X-coordinate of the hexagon center
- `y`: Y-coordinate of the hexagon center
- `color`: Filling color of the hexagon (default NA)
- `border`: Border color of the hexagon (default "gray")
- `unitcell`: The distance side to side between two parallel sides of the hexagon (default 1)

**Value**

This function draws an hexagon on a plot

**Author(s)**

Sabina Licen

Hexagons

*Function to draw an hexagonal SOM map*

**Description**

Draws an hexagonal SOM map using x, y coordinates for the hexagon centers

**Usage**

`Hexagons(Coords, Row, Col, color = NA, border = "gray", unitcell = 1)`

**Arguments**

- `Coords`: matrix containing the x and y coordinates of the hexagon centers
- `Row`: Number of SOM map rows
- `Col`: Number of SOM map columns
- `color`: Filling color of the hexagons (default NA)
- `border`: Border color of the hexagons (default "gray")
- `unitcell`: The distance side to side between two parallel sides of the hexagon (default 1)
Value

A hexagonal SOM map

Author(s)

Sabina Licen

References


Examples

```r
Coord<-CodeCoord(10,5)
Hexagons(Coord,10,5)
```

Description

Generates a SOM map colored according to cluster splitting

Usage

```r
HexagonsClus(
    Centroids,
    Cluster,
    BCentr,
    Coord,
    Row,
    Col,
    colSeq = rainbow(nrow(Centroids))
)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centroids</td>
<td>Centroids matrix</td>
</tr>
<tr>
<td>Cluster</td>
<td>Vector containing cluster number assignment for prototypes</td>
</tr>
<tr>
<td>BCentr</td>
<td>Best Matching Unit of the cluster centroids</td>
</tr>
<tr>
<td>Coord</td>
<td>Prototype coordinates for plotting the map</td>
</tr>
<tr>
<td>Row</td>
<td>Number of SOM map rows</td>
</tr>
<tr>
<td>Col</td>
<td>Number of SOM map columns</td>
</tr>
<tr>
<td>colSeq</td>
<td>Color sequence for the clusters</td>
</tr>
</tbody>
</table>
HexagonsVar

Value
A SOM map colored according to cluster splitting

Author(s)
Sabina Licen

References

Description
Multiple plots that show the distribution of the modeled variables on the SOM map

Usage
HexagonsVar(Dms, codebook, Coords, Row, Col)

Arguments
Dms A vector of length 2, where the first argument specifies the number of rows and the second the number of columns of plots (see mfrow in par)
codebook SOM codebook
Coords Prototype coordinates for plotting the map
Row Number of SOM map rows
Col Number of SOM map columns

Details
The function plots a SOM map for the values of each modeled variable using a grayscale according to quartiles, from white (lower outliers), followed by grayscale (quartiles) and black (upper outliers). The outliers and quartiles are evaluated by boxplot function applying default parameters.

Value
SOM map plots for the values of each modeled variable using a grayscale according to quartiles

Author(s)
Sabina Licen
References


See Also

boxplot, par

---

**HexaHits**

**Hits distribution on the SOM map**

Description

Plot a SOM map with filled hexagons according to the number of hits

Usage

```r
HexaHits(hits, Coord, Row, Col, color = "black")
```

Arguments

- `hits`: Vector with number of hits for each prototype
- `Coord`: Prototype coordinates for plotting the map
- `Row`: Number of SOM map rows
- `Col`: Number of SOM map columns
- `color`: color filling of the hexagons

Value

Plot a SOM map with filled hexagons according to the number of hits

Author(s)

Sabina Licen

References

HexaHitsQuant

Hits distribution on the SOM map

Description

Plot a SOM map with hits plotted as grayscale according to quartiles

Usage

HexaHitsQuant(hits, Coord, Row, Col)

Arguments

- hits: Vector with number of hits for each prototype
- Coord: Prototype coordinates for plotting the map
- Row: Number of SOM map rows
- Col: Number of SOM map columns

Details

The function plots a SOM map with hits represented as grayscale according to quartiles, from white (lower outliers) followed by grayscale (quartiles) and black (upper outliers). The prototype with the maximum number of hits is represented by a red hexagon. The outliers and quartiles are evaluated by boxplot function applying default parameters.

Value

Plot a SOM map with hits represented as grayscale according to quartiles

Author(s)

Sabina Licen

References


See Also

boxplot
HexaQerrs

Relative quantization error distribution on the SOM map

Description

Plot a SOM map with relative quantization error plotted as grayscale according to quartiles

Usage

HexaQerrs(bmus, qerrs, Coord, Row, Col, color = "black")

Arguments

bmus  Vector with Best Matching Unit for each experimental sample
qerrs  Vector with quantization error for each experimental sample
Coord  Prototype coordinates for plotting the map
Row    Number of SOM map rows
Col    Number of SOM map columns
color  color filling of the hexagons

Details

The function evaluate the relative quantization error for each prototype dividing the sum of quantization errors for experimental samples represented by the single prototype by the number of hits of the same prototype, then plots a SOM map with filled hexagons according to the relative quantization error.

Value

Plot a SOM map with filled hexagons according to the relative quantization error

Author(s)

Sabina Licen

References

HexaQerrsQuant

Realtime quantization error distribution on the SOM map

Description
Plot a SOM map with relative quantization error plotted as grayscale according to quartiles

Usage
HexaQerrsQuant(bmus, qerrs, Coord, Row, Col)

Arguments
- **bmus**: Vector with Best Matching Unit for each experimental sample
- **qerrs**: Vector with quantization error for each experimental sample
- **Coord**: Prototype coordinates for plotting the map
- **Row**: Number of SOM map rows
- **Col**: Number of SOM map columns

Details
The function evaluates the relative quantization error for each prototype dividing the sum of quantization errors for experimental sample represented by the single prototype by the number of hits of the same prototype, then plots a SOM map with the relative quantization error represented as grayscale according to quartiles, from white (lower outliers) followed by grayscale (quartiles) and black (upper outliers). The outliers and quartiles are evaluated by boxplot function applying default parameters.

Value
Plot a SOM map with relative quantization error represented as grayscale according to quartiles

Author(s)
S. Licen

References

See Also
boxplot
**kmeans_clustersRProg**  
*K-means algorithm applied for different values of clusters*

**Description**

The `som_kmeansR` function with 100 epochs training is run for a custom number of times for each k value of clusters and the best of these is selected based on sum of squared errors (err). The Davies-Bouldin index is calculated for each k-clustering. The function has been coded in R code starting from `kmeans_clusters.m` script present in somtoolbox for Matlab by Vesanto and adapted to show a progress bar when working embedded in the shiny app.

**Usage**

```r
kmeans_clustersRProg(codebook, k = 5, times = 5, seed = NULL)
```

**Arguments**

- `codebook`  
  SOM codebook
- `k`  
  Maximum number of clusters (the function will be run from 2 to k clusters)
- `times`  
  Number of times the `som_kmeansR` function is iterated
- `seed`  
  Number for `set.seed` function

**Value**

This function returns a list containing the cluster number assignment for each sample, the cluster centroids, the total quantization error, the DB-index for each number of clusters, and the random seed number used

**Author(s)**

Sabina Licen, Pierluigi Barbieri

**References**


**See Also**

`som_mdistR`, `som_kmeansRProg`, `db_indexR`
NClusChange  \hspace{1cm} Custom number sequence for clusters

**Description**

Changes the input vector according the custom number sequence for clusters

**Usage**

\[ \text{NClusChange}(\text{Vector, Centroids, NCh}) \]

**Arguments**

- **Vector**: Vector containing cluster number assignment for prototypes or experimental data
- **Centroids**: Centroids matrix
- **NCh**: Vector with custom sequence of numbers for clusters

**Value**

A vector of same length as input vector with cluster numbers changed according to custom input

**Author(s)**

Sabina Licen

---

**paramQuant**  \hspace{1cm} Basic statistics of values present in the input vector

**Description**

Generate basic statistics for the input vector

**Usage**

\[ \text{paramQuant}(\text{param}) \]

**Arguments**

- **param**: Numeric vector

**Details**

The outliers and quartiles are evaluated by boxplot function applying default parameters.
Description
The function starts the SOMEnv GUI

Usage
SomEnvGUI()

Value
This function starts the graphical user interface with the default system browser. The main help suggestion for using the tool are embedded in the GUI

Author(s)
Sabina Licen, Marco Franzon, Tommaso Rodani

References

Examples
## Not run:
SomEnvGUI()

## End(Not run)
**SOMtopol**

*Topographical error for the SOM map*

**Description**

Calculate topographical error for the SOM map

**Usage**

SOMtopol(dataset, codebook, grid)

**Arguments**

- **dataset**: Experimental data used for training the map
- **codebook**: SOM codebook
- **grid**: SOM grid expressed as a matrix of x and y coordinates of the map units

**Value**

This function returns the topographical error

**Author(s)**

Sabina Licen

**References**


---

**som_dimR**

*Calculate map dimensions*

**Description**

Generate SOM map dimensions according to Vesanto heuristic rules based on the first two eigenvalues of the experimental data and their related eigenvectors. The function has been coded in R code starting from som_dim.m script present in somtoolbox for Matlab by Vesanto and adapted for the use in the shiny app

**Usage**

som_dimR(dataset, type = "regular")
som_initR

Calculate initialization matrix for SOM training

Description

Generate SOM map initialization matrix according to Vesanto heuristic rules related to map dimensions, the first two eigenvalues of the experimental data and their related eigenvectors. The function has been coded in R code starting from som_init.m script present in somtoolbox for Matlab by Vesanto and adapted for the use in the shiny app.

Usage

som_initR(dataset, Row, Col, munits)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataset</td>
<td>Experimental data</td>
</tr>
<tr>
<td>Row</td>
<td>Number of SOM map rows</td>
</tr>
<tr>
<td>Col</td>
<td>Number of SOM map columns</td>
</tr>
<tr>
<td>munits</td>
<td>Number of SOM map units (Row*Col)</td>
</tr>
</tbody>
</table>

Author(s)

Sabina Licen, Pierluigi Barbieri

References


See Also
eigen, cor

Examples

library(datasets)
som_dimR(iris[,1:4], type="small")
**som_kmeansRProg**

**Value**
This function returns an initialization matrix for SOM training

**Author(s)**
Sabina Licen, Pierluigi Barbieri

**References**

**Examples**
```r
SOMdim<-som_dimR(iris[,1:4], type="small")
SOMinit<-som_initR(iris[,1:4],SOMdim$Row,SOMdim$Col,SOMdim$munits)
```

---

**som_kmeansRProg**

*K-means algorithm applied for a specific number of clusters*

**Description**
The training is run for a custom number of epochs for k number of clusters

**Usage**
```
som_kmeansRProg(codebook, k, epochs, seed = NULL)
```

**Arguments**
- `codebook` SOM codebook
- `k` Number of clusters
- `epochs` Number of training epochs
- `seed` Number for set.seed function

**Details**
The function has been coded in R code starting from som_kmeans.m script present in somtoolbox for Matlab by Vesanto and adapted to show a progress bar when working embedded in the shiny app.

**Value**
This function returns a list containing the cluster number assignment for each sample, the cluster centroids, the total quantization error, and the random seed number used
**Author(s)**
Sabina Licen, Pierluigi Barbieri

**References**

**See Also**
set.seed

<table>
<thead>
<tr>
<th>som_mdistR</th>
<th>Evaluate pairwise distance matrix for the given codebook</th>
</tr>
</thead>
</table>

**Description**
The function has been coded in R code starting from som_mdist.m script present in somtoolbox for Matlab by Vesanto and adapted for the use in the shiny app

**Usage**
som_mdistR(codebook)

**Arguments**
codebook SOM codebook

**Value**
Distance matrix

**Author(s)**
Sabina Licen, Pierluigi Barbieri

**References**

**See Also**
db_indexR
**som_umatR**

*Unified distance matrix for the SOM map*

**Description**

The function has been coded in R code starting from `som_umat.m` script present in `somtoolbox` for Matlab by Vesanto and adapted for the use in the shiny app.

**Usage**

```r
som_umatR(codebook, Row, Col)
```

**Arguments**

- `codebook` : SOM codebook
- `Row` : Number of SOM map rows
- `Col` : Number of SOM map columns

**Value**

The unified distance matrix for the SOM map

**Author(s)**

Sabina Licen, Pierluigi Barbieri

**References**


---

**UmatGraph**

*U-matrix plot*

**Description**

Plot of Unified Distance Matrix using a colored scale according to quartiles

**Usage**

```r
UmatGraph(umat, Row, Col, colorscale = c("bw", "gs"))
```
Arguments

umat  Unified Distance Matrix
Row     Number of SOM map rows
Col     Number of SOM map columns
colorscale  Either "bw" for grayscale or "gs" for green-white scale

Details

The function plots a U-matrix map for the values of each modeled variable using a grayscale according to quartiles, from darker color (lower distances) to lighter color (higher distances). The quartiles are evaluated by boxplot function applying default parameters.

Value

Plot of Unified Distance Matrix using a grayscale or (green-white scale) according to quartiles

Author(s)

Sabina Licen

References


See Also

boxplot, som_umatR
Index

BmusCentr, 2
BmusClus, 3
BoxClus, 4
BoxUnits, 5

ClusCol, 6
CodeCoord, 6

DailyBar, 7
db_indexR, 8

Freq, 9
FreqD, 9
FreqM, 10

Hexa, 11
Hexagons, 11
HexagonsClus, 12
HexagonsVar, 13
HexaHits, 14
HexaHitsQuant, 15
HexaQerrs, 16
HexaQerrsQuant, 17

kmeans_clustersRProg, 18

NClusChange, 19

paramQuant, 19

som_dimR, 21
som_initR, 22
som_kmeansRProg, 23
som_mdistR, 24
som_umatR, 25
SomEnvGUI, 20
SOMtopol, 21

UmatGraph, 25