Package 'Umatrix'

May 28, 2025

Type Package

Title Visualization of Structures in High-Dimensional Data

Version 4.0.2

Date 2025-05-28

Description By gaining the property of emergence through self-organization, the enhancement of SOMs(self organizing maps) is called Emergent SOM (ESOM). The result of the projection by ESOM is a grid of neurons which can be visualised as a three dimensional landscape in form of the Umatrix. Further details can be found in the referenced publications (see url). This package offers tools for calculating and visualising the ESOM as well as Umatrix, Pmatrix and UStarMatrix. All the functionality is also available through graphical user interfaces implemented in 'shiny'. Based on the recognized data structures, the method can be used to generate new data.

Imports Rcpp, ggplot2, shiny, shinyjs, reshape2, fields, plyr, png, tools, grid, abind, deldir, geometry, pdist, AdaptGauss, DataVisualizations, ggrepel

Suggests rgl

LinkingTo Rcpp

Depends R (>= 3.5.0)

URL http://wscg.zcu.cz/wscg2016/short/A43-full.pdf

NeedsCompilation yes

Author Florian Lerch [aut],
Michael Thrun [aut],
Felix Pape [ctb],
Jorn Lotsch [aut, cre],
Raphael Paebst [ctb],
Alfred Ultsch [aut]

Maintainer Jorn Lotsch <j.lotsch@em.uni-frankfurt.de>

Repository CRAN

Date/Publication 2025-05-28 12:40:02 UTC

Contents

Umatrix-package 2
BMUHepta 3
calculate_Delauny_radius 4
esomTrain 5
generate_data 6
Hepta 7
iClassification 8
iEsomTrain 9
iUmapIsland 10
iUstarmatrix 10
plotMatrix 11
pmatrixForEsom 13
showMatrix3D 14
umatrixForEsom 15
ustarmatrixCalc 15

Index **17**

`Umatrix-package` *Umatrix-package*

Description

The **ESOM**(emergent self organizing map) is an improvement of the regular **SOM**(self organizing map) which allows for toroid grids of neurons and is intended to be used in combination with the **Umatrix**. The set of neurons is referred to as **weights** within this package, as they represent the values within the high dimensional space. The neuron with smallest distance to a datapoint is called a **Bestmatch** and can be considered as projection of said datapoint. As the Umatrix is usually toroid, it is drawn four consecutive times to remove border effects. An island, or Imx, is a filter mask, which cuts out a subset of the Umatrix, which shows every point only a single time while avoiding border effects cutting through potential clusters. Finally the Pmatrix shows the density structures within the grid, by a set radius. It can be combined with the Umatrix resulting in the UStarMatrix, which is therefore a combination of density based structures as well as clearly divided ones.

References

Ultsch, A.: Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series, In Oja, E. & Kaski, S. (Eds.), Kohonen maps, (1 ed., pp. 33-46), Elsevier, 1999.

Ultsch, A.: Maps for the visualization of high-dimensional data spaces, Proc. Workshop on Self organizing Maps (WSOM), pp. 225-230, Kyushu, Japan, 2003.

Ultsch, A.: U* C: Self-organized Clustering with Emergent Feature Maps, Lernen, Wissensentdeckung und Adaptivitaet (LWA), pp. 240-244, Saarbruecken, Germany, 2005.

Lotsch, J., Ultsch, A.: Exploiting the Structures of the U-Matrix, in Villmann, T., Schleif, F.-M., Kaden, M. & Lange, M. (eds.), Proc. Advances in Self-Organizing Maps and Learning Vector Quantization, pp. 249-257, Springer International Publishing, Mittweida, Germany, 2014.

Ultsch, A., Behnisch, M., Lotsch, J.: ESOM Visualizations for Quality Assessment in Clustering, In Merenyi, E., Mendenhall, J. M. & O'Driscoll, P. (Eds.), Advances in Self-Organizing Maps and Learning Vector Quantization: Proceedings of the 11th International Workshop WSOM 2016, pp. 39-48, Houston, Texas, USA, January 6-8, 2016, (10.1007/978-3-319-28518-4_3), Cham, Springer International Publishing, 2016.

Thrun, M. C., Lerch, F., Lotsch, J., Ultsch, A.: Visualization and 3D Printing of Multivariate Data of Biomarkers, in Skala, V. (Ed.), International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision,Plzen, 2016.

`BMUHepta` *Best matching units (BMU) of Hepta from FCPS (Fundamental Clustering Problem Suite)*

Description

Best matching units (BMU) of an ESOM projection of the Hepta data set from FCPS (Fundamental Clustering Problem Suite) on an 80 x 40 planar grid of artifical neurons.

Usage

```
data("BMUHepta")
```

Details

Size 212, Dimensions 3 (key, linecoordinates, columncoorditaes)

Classes 7, stored in `Hepta$Cls`

References

Ultsch A, Lotsch J: Machine-learned cluster identification in high-dimensional data. J Biomed Inform. 2017 Feb;66:95-104. doi: 10.1016/j.jbi.2016.12.011. Epub 2016 Dec 28.

Examples

```
data("BMUHepta")
str("BMUHepta")
```

`calculate_Delauny_radius`
Calculate the Delauny graph based radius

Description

Function to calculate the radius for data generation.

Usage

```
calculate_Delauny_radius(Data, BestMatches,
Columns = 80,  Lines = 50,  Toroid = TRUE)
```

Arguments

`Data`	Matrix of data (as submitted to Umatrix generation)
`BestMatches`	Array with positions of Bestmatches
`Columns`	Number of columns of the Umatrix
`Lines`	Number of columns of the Umatrix
`Toroid`	Whether a toroid Umatrx was used

Value

Returns a list of results.

`neighbourDistances`	Distances on the Umatrix neigborhood matrix.
`RadiusByEM`	Radius suggested by EM algorithm.

References

Ultsch A, Lotsch J: Machine-learned cluster identification in high-dimensional data. J Biomed Inform. 2017 Feb;66:95-104. doi: 10.1016/j.jbi.2016.12.011. Epub 2016 Dec 28.

Examples

```
## Not run:
data("Hepta")
data("HeptaBMU")
DelaunyHepta <- calculate_Delauny_radius(Data = Hepta$Data, BestMatches = HeptaBMU, Toroid = FALSE)

## End(Not run)
```

esomTrain *Train an ESOM (emergent self organizing map) and project data*

Description

The ESOM (emergent self organizing map) algorithm as defined by [Ultsch 1999]. A set of weigths(neurons) on a two-dimensional grid get trained to adapt the given datastructure. The weights will be used to project data on a two-dimensional space, by seeking the BestMatches for every datapoint.

Arguments

`Data`	Data that will be used for training and projection
`Lines`	Height of grid
`Columns`	Width of grid
`Epochs`	Number of Epochs the ESOM will run
`Toroid`	If TRUE, the grid will be toroid
`NeighbourhoodFunction`	Type of Neighbourhood; Possible values are: "cone", "mexicanhat" and "gauss"
`StartLearningRate`	Initial value for LearningRate
`EndLearningRate`	Final value for LearningRate
`StartRadius`	Start value for the Radius in which will be searched for neighbours
`EndRadius`	End value for the Radius in which will be searched for neighbours
`NeighbourhoodCooling`	Cooling method for radius; "linear" is the only available option at the moment
`LearningRateCooling`	Cooling method for LearningRate; "linear" is the only available option at the moment
`shinyProgress`	Generate progress output for shiny if Progress Object is given
`ShiftToHighestDensity`	If True, the Umatrix will be shifted so that the point with highest density will be at the center
`InitMethod`	name of the method that will be used to choose initializations Valid Inputs: "uni_min_max": uniform distribution with minimum and maximum from sampleData "norm_mean_std": normal distribuation based on mean and standard deviation of sampleData
`Key`	Vector of numeric keys matching the datapoints. Will be added to Bestmatches
`UmatrixForEsom`	If TRUE, Umatrix based on resulting ESOM is calculated and returned

Details

On a toroid grid, opposing borders are connected.

Value

List with

`BestMatches`	BestMatches of datapoints
`Weights`	Trained weights
`Lines`	Height of grid
`Columns`	Width of grid
`Toroid`	TRUE if grid is a toroid
`JumpingDataPointsHist`	Nr of DataPoints that jumped to a different BestMatch in every epoch

References

Kohonen, T., Self-organized formation of topologically correct feature maps. Biological cybernetics, 1982. 43(1): p. 59-69.

Ultsch, A., Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series. Kohonen maps, 1999. 46: p. 33-46.

Examples

```
data('Hepta')
res=esomTrain(Hepta$Data, Key = 1:nrow(Hepta$Data))
```

`generate_data` *Generative ESOM*

Description

Function to generate new data with the same structure as the input data.

Usage

```
generate_data(Data, density_radius, Cls = NULL, gen_per_data = 10)
```

Arguments

`Data`	Matrix of data (as submitted to Umatrix generation)
`density_radius`	Numeric value of data generation radius
`Cls`	Classification of the data as a vector
`gen_per_data`	New isntances per original iunstance to be generated

Value

Returns a list of results.

`original_data` The input data.

`original_classes`
The input classes.

`generated_data` The generated data.

`generated_classes`
The generated classes.

References

Ultsch A, Lotsch J: Machine-learned cluster identification in high-dimensional data. J Biomed Inform. 2017 Feb;66:95-104. doi: 10.1016/j.jbi.2016.12.011. Epub 2016 Dec 28.

Examples

```
## Not run:
data("Hepta")
data("HeptaBMU")
HeptaData <- Hepta$Data
HeptaCls <y- Hepta$Cls
HeptaGenerated <- generate_data(HeptaData, 1, HeptaCls )

## End(Not run)
```

`Hepta` *Hepta from FCPS (Fundamental Clustering Problem Suite)*

Description

Dataset with 7 easily seperable classes.

Usage

`data("Hepta")`

Details

Size 212, Dimensions 3, stored in `Hepta$Data`

Classes 7, stored in `Hepta$Cls`

References

Ultsch, A.: U* C: Self-organized Clustering with Emergent Feature Maps, Lernen, Wissensentdeckung und Adaptivitaet (LWA), pp. 240-244, Saarbruecken, Germany, 2005.

Examples

```
data("Hepta")
str("Hepta")
```

---

iClassification *GUI for manual classification*

Description

This tool is a 'shiny' GUI that visualizes a given Umatrix and allows the user to select areas and mark them as clusters.

Arguments

`Umatrix`	Matrix of Umatrix Heights
`BestMatches`	Array with positions of Bestmatches
`Cls`	Classification of the Bestmatches
`Imx`	Matrix of an island that will be cut out of the Umatrix
`Toroid`	Are BestMatches placed on a toroid grid? TRUE by default

Value

A vector containing the selected class ids. The order is corresponding to the given Bestmatches

References

Thrun, M. C., Lerch, F., Loetsch, J., Ultsch, A.: Visualization and 3D Printing of Multivariate Data of Biomarkers, in Skala, V. (Ed.), International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision,Plzen, 2016.

Examples

```
## Not run:
data("Hepta")
e = esomTrain(Hepta$Data, Key = 1:nrow(Hepta$Data))
cls = iClassification(e$Umatrix, e$BestMatches)

## End(Not run)
```

`iEsomTrain` *iEsomTrain*

Description

Trains the ESOM and shows the Umatrix.

Arguments

`Data`	Matrix of Data that will be used to learn. One DataPoint per row
`BestMatches`	Array with positions of Bestmatches
`Cls`	Classification of the Bestmatches as a vector
`Key`	Numeric vector of keys matching the Bestmatches
`Toroid`	Are BestMatches placed on a toroid grid? TRUE by default

Value

List with

`Umatrix`	matrix with height values of the umatrix
`BestMatches`	matrix containing the bestmatches
`Lines`	number of lines of the chosen ESOM
`Columns`	number of columns of the chosen ESOM
`Epochs`	number of epochs of the chosen ESOM
`Weights`	List of weights
`Toroid`	True if a toroid grid was used
`EsomDetails`	Further details describing the chosen ESOM parameters
`JumpingDataPointsHist`	Number of Datapoints that jumped to another neuron in each epoch

References

Thrun, M. C., Lerch, F., Loetsch, J., Ultsch, A.: Visualization and 3D Printing of Multivariate Data of Biomarkers, in Skala, V. (Ed.), International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision,Plzen, 2016.

`iUmapIsland` *iUmapIsland*

Description

The toroid Umatrix is usually drawn 4 times, so that connected areas on borders can be seen as a whole. An island is a manual cutout of such a tiled visualization, that is selected such that all connected areas stay intact. This 'shiny' tool allows the user to do this manually.

Arguments

`Umatrix`	Matrix of Umatrix Heights
`BestMatches`	Array with positions of BestMatches
`Cls`	Classification of the BestMatches

Value

Boolean Matrix that represents the island within the tiled Umatrix

References

Thrun, M. C., Lerch, F., Loetsch, J., Ultsch, A.: Visualization and 3D Printing of Multivariate Data of Biomarkers, in Skala, V. (Ed.), International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision,Plzen, 2016.

Examples

```
## Not run:
data("Hepta")
e = esomTrain(Hepta$Data, Key = 1:nrow(Hepta$Data))
Imx = iUmapIsland(e$Umatrix, e$BestMatches)
plotMatrix(e$Umatrix, e$BestMatches, Imx = Imx$Imx)

## End(Not run)
```

`iUstarmatrix` *iUstarmatrix*

Description

Calculates the Ustarmatrix by combining a Umatrix with a Pmatrix.

Arguments

`Weights`	Weights that were trained by the ESOM algorithm
`Lines`	Height of the used grid
`Columns`	Width of the used grid
`Data`	Matrix of Data that was used to train the ESOM. One datapoint per row
`Imx`	Island mask that will be cut out from displayed Umatrix
`Cls`	Classification of the Bestmatches
`Toroid`	Are weights placed on a toroid grid?

Value

`Ustarmatrix`	matrix with height values of the Ustarmatrix

References

Thrun, M. C., Lerch, F., Loetsch, J., Ultsch, A.: Visualization and 3D Printing of Multivariate Data of Biomarkers, in Skala, V. (Ed.), International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision,Plzen, 2016.

`plotMatrix` *plotMatrix*

Description

Draws a plot based of given Umatrix or Pmatrix.

Arguments

`Matrix`	Umatrix or Pmatrix to be plotted
`BestMatches`	Positions of BestmMtches to be plotted onto the Umatrix
`Cls`	Class identifier for the BestMatches
`ClsColors`	Vector of colors that will be used to colorize the different classes
`ColorStyle`	If "Umatrix" the colors of a Umatrix (Blue -> Green -> Brown -> White) will be used; If "Pmatrix" the colors of a Pmatrix (White -> Yellow -> Red) will be used
`Toroid`	Should the Umatrix be drawn 4times?
`BmSize`	Integer between 0.1 and 5, magnification factor of the drawn BestMatch circles
`DrawLegend`	If TRUE, a color legend will be drawn next to the plot
`FixedRatio`	If TRUE, the plot will be drawn with a fixed ratio of x and y axis
`CutoutPol`	Only draws the area within given polygon
`Nrlevels`	Number of height levels that will be used within the Umatrix

`TransparentContours`
: Use half transparent contours. Looks better but is slow

`Imx`
: Mask to cut out an island. Every value should be either 1 (stays in) or 0 (gets cut out)

`Clean`
: If TRUE axis, margins, ... surrounding the Umatrix image will be removed

`RemoveOcean`
: If TRUE, the surrounding blue area around an island will be reduced as much as possible (while still maintaining a rectangular form)

`TransparentOcean`
: If TRUE, the surrounding blue area around an island will be transparent

`Title`
: A title that will be drawn above the plot

`BestMatchesLabels`
: Vector of strings corresponding to the order of BestMatches which will be drawn on the plot as labels

`BestMatchesShape`
: Numeric value of Shape that will be used. Responds to the usual shapes of ggplot

`MarkDuplicatedBestMatches`
: If TRUE, BestMatches that are shown more than once within an island, will be marked

`YellowCircle`
: If TRUE, a yellow circle is drawn around Bestmatches to distinct them better from background

Details

The heightScale (nrlevels) is set at the proportion of the 1 percent quantile against the 99 percent quantile of the matrix values.

Value

A 'ggplot' of a Matrix

References

Thrun, M. C., Lerch, F., Loetsch, J., Ultsch, A.: Visualization and 3D Printing of Multivariate Data of Biomarkers, in Skala, V. (Ed.), International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision,Plzen, 2016.

Ultsch, A.: Maps for the visualization of high-dimensional data spaces, Proc. Workshop on Self organizing Maps (WSOM), pp. 225-230, Kyushu, Japan, 2003.

Siemon, H.P., Ultsch,A.: Kohonen Networks on Transputers: Implementation and Animation, in: Proceedings Intern. Neural Networks, Kluwer Academic Press, Paris, pp. 643-646, 1990.

Examples

```
data("Hepta")
e = esomTrain(Hepta$Data, Key = 1:nrow(Hepta$Data))
plotMatrix(e$Umatrix,e$BestMatches)
```

pmatrixForEsom *pmatrixForEsom*

Description

Generates a Pmatrix based on the weights of an ESOM.

Arguments

Data	A [n,k] matrix containing the data
Weights	Weights stored as a list in a 2D matrix
Lines	Number of lines of the SOM that is described by weights
Columns	Number of columns of the SOM that is described by weights
Radius	The radius for measuring the density within the hypersphere
PlotIt	If set the Pmatrix will also be plotted
Toroid	Are BestMatches placed on a toroid grid? TRUE by default

Value

UstarMatrix

References

Ultsch, A.: Maps for the visualization of high-dimensional data spaces, Proc. Workshop on Self organizing Maps (WSOM), pp. 225-230, Kyushu, Japan, 2003.

Ultsch, A., Loetsch, J.: Computed ABC Analysis for Rational Selection of Most Informative Variables in Multivariate Data, PloS one, Vol. 10(6), pp. e0129767. doi 10.1371/journal.pone.0129767, 2015.

Thrun, M. C., Lerch, F., Loetsch, J., Ultsch, A.: Visualization and 3D Printing of Multivariate Data of Biomarkers, in Skala, V. (Ed.), International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision,Plzen, 2016.

Examples

```
data("Hepta")
e = esomTrain(Hepta$Data, Key = 1:nrow(Hepta$Data))
Pmatrix = pmatrixForEsom(Hepta$Data,
                         e$Weights,
                         e$Lines,
                         e$Columns,
                         e$Toroid)
plotMatrix(Pmatrix, ColorStyle = "Pmatrix")
```

showMatrix3D *showMatrix3D*

Description

Visualizes the matrix(Umatrix/Pmatrix) in an interactive window in 3D.

Arguments

`Matrix`	Matrix to be plotted
`BestMatches`	Positions of BestMatches to be plotted onto the matrix
`Cls`	Class identifier for the BestMatch at the given point
`Imx`	a mask (island) that will be used to cut out the Umatrix
`Toroid`	Should the Matrix be drawn 4 times (in a toroid view)
`HeightScale`	Optional. Scaling Factor for Mountain Height
`BmSize`	Size of drawn BestMatches
`RemoveOcean`	Remove as much area sourrounding an island as possible
`ColorStyle`	Either "Umatrix" or "Pmatrix" respectevily for their colors
`ShowAxis`	Draw an axis arround the drawn matrix
`SmoothSlope`	Try to increase the island size, to get smooth slopes around the island
`ClsColors`	Vector of colors that will be used for classes
`FileName`	Name for a stl file to write the Matrix to

Details

The heightScale is set at the proportion of the 1 percent quantile against the 99 percent quantile of the Matrix values.

References

Thrun, M. C., Lerch, F., Loetsch, J., Ultsch, A.: Visualization and 3D Printing of Multivariate Data of Biomarkers, in Skala, V. (Ed.), International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision,Plzen, 2016.

Examples

```
## Not run:
data("Hepta")
e = esomTrain(Hepta$Data, Key = 1:nrow(Hepta$Data))
showMatrix3D(e$Umatrix)

## End(Not run)
```

umatrixForEsom *umatrixForEsom*

Description

Calculate the Umatrix for given ESOM projection

Arguments

`Weights`	Weights from which the Umatrix will be calculated
`Lines`	Number of lines of the SOM that is described by weights
`Columns`	Number of columns of the SOM that is described by weights
`Toroid`	Boolean describing if the neural grid should be borderless

Value

Umatrix

References

Ultsch, A. and H.P. Siemon, Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis. 1990.

Examples

```
data("Hepta")
e = esomTrain(Hepta$Data, Key = 1:nrow(Hepta$Data))
umatrix = umatrixForEsom(e$Weights,
                         Lines=e$Lines,
                         Columns=e$Columns,
                         Toroid=e$Toroid)
plotMatrix(umatrix,e$BestMatches)
```

ustarmatrixCalc *ustarmatrixCalc*

Description

The UStarMatrix is a combination of the Umatrix (average distance to neighbours) and Pmatrix (density in a point). It can be used to improve the Umatrix, if the dataset contains density based structures.

Arguments

`Umatrix`	A given Umatrix
`Pmatrix`	A density matrix

Value

UStarMatrix

References

Ultsch, A. U* C: Self-organized Clustering with Emergent Feature Maps. in Lernen, Wissensentdeckung und Adaptivitaet (LWA). 2005. Saarbruecken, Germany.

Examples

```
data("Hepta")
e = esomTrain(Hepta$Data, Key = 1:nrow(Hepta$Data))
Pmatrix = pmatrixForEsom(Hepta$Data,
                         e$Weights,
                         e$Lines,
                         e$Columns,
                          e$Toroid)
Ustarmatrix = ustarmatrixCalc(e$Umatrix, Pmatrix)
plotMatrix(Ustarmatrix, e$BestMatches)
```

Index

∗ **ESOM**
esomTrain, 5
Umatrix-package, 2
∗ **High-Dimensional Data**
Umatrix-package, 2
∗ **High-dimensional**
Umatrix-package, 2
∗ **Highdimensional**
Umatrix-package, 2
∗ **P-Matrix**
pmatrixForEsom, 13
∗ **P-matrix**
pmatrixForEsom, 13
∗ **Pmatrix**
pmatrixForEsom, 13
∗ **Projection**
Umatrix-package, 2
∗ **SOM**
esomTrain, 5
Umatrix-package, 2
∗ **U*-Matrix**
ustarmatrixCalc, 15
∗ **U*-matrix**
ustarmatrixCalc, 15
∗ **U*matrix**
ustarmatrixCalc, 15
∗ **U-Matrix**
Umatrix-package, 2
umatrixForEsom, 15
∗ **U-matrix**
umatrixForEsom, 15
∗ **Umatrix**
Umatrix-package, 2
umatrixForEsom, 15
∗ **Ustar-Matrix**
ustarmatrixCalc, 15
∗ **Ustar-matrix**
ustarmatrixCalc, 15
∗ **Ustarmatrix**
ustarmatrixCalc, 15
∗ **Visualization**
Umatrix-package, 2
∗ **datasets**
BMUHepta, 3
Hepta, 7
∗ **esom**
esomTrain, 5
∗ **self organizing map**
esomTrain, 5
∗ **som**
esomTrain, 5

BMUHepta, 3

calculate_Delauny_radius, 4

esomTrain, 5

generate_data, 6

Hepta, 7

iClassification, 8
iEsomTrain, 9
iUmapIsland, 10
iUstarmatrix, 10

plotMatrix, 11
pmatrixForEsom, 13

showMatrix3D, 14

Umatrix-package, 2
umatrixForEsom, 15
ustarmatrixCalc, 15