Package 'KODAMA'

November 5, 2024

Version 2.4.1

Date 2023-01-11

Maintainer Stefano Cacciatore <tkcaccia@gmail.com>

Title Knowledge Discovery by Accuracy Maximization

Description An unsupervised and semi-supervised learning algorithm that performs feature extraction from noisy and high-
dimensional data. It facilitates identification of patterns representing underlying
groups on all samples in a data set. Based on Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA.
(2017) Bioinformatics <doi:10.1093/bioinformatics/btw705> and Cacciatore S, Luchinat C, Tenori L. (2014)
Proc Natl Acad Sci USA <doi:10.1073/pnas.1220873111>.

Depends R (>= 2.10.0), stats, minerva, Rtsne, umap

Imports Rcpp (>= 0.12.4)

LinkingTo Rcpp, RcppArmadillo

Suggests rgl, knitr, rmarkdown

VignetteBuilder knitr

SuggestsNote No suggestions

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Author Stefano Cacciatore [aut, trl, cre]
(<https://orcid.org/0000-0001-7052-7156>),
Leonardo Tenori [aut] (<https://orcid.org/0000-0001-6438-059X>)

Date/Publication 2024-11-05 05:38:47 UTC

Contents

`categorical.test` *Categorical Information*

Description

Summarization of the categorical information.

Usage

```
categorical.test (name,x,y,total.column=FALSE,...)
```

Arguments

`name`	the name of the feature.
`x`	the information to summarize.
`y`	the classification of the cohort.
`total.column`	option to visualize the total (by default = "FALSE").
`...`	further arguments to be passed to the function.

Value

The function returns a table with the summarized information and The p-value computated using the Fisher's test.

Author(s)

Stefano Cacciatore

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

See Also

correlation.test,continuous.test, txtsummary

Examples

```
data(clinical)

hosp=clinical[,"Hospital"]
gender=clinical[,"Gender"]
GS=clinical[,"Gleason score"]
BMI=clinical[,"BMI"]
age=clinical[,"Age"]

A=categorical.test("Gender",gender,hosp)
B=categorical.test("Gleason score",GS,hosp)

C=continuous.test("BMI",BMI,hosp,digits=2)
D=continuous.test("Age",age,hosp,digits=1)

rbind(A,B,C,D)
```

---

clinical *Clinical Data of a Cohort of Prostate Cancer Patiens*

Description

The data belong to a cohort of 35 patients with prostate cancer from two different hospitals.

Usage

```
data(clinical)
```

Value

The data.frame "prcomp" with the following elements: "Hospital", "Gender", "Gleason score", "BMI", and "Age".

Examples

```
data(clinical)

head(clinical)
```

`continuous.test` *Continuous Information*

Description

Summarization of the continuous information.

Usage

```
continuous.test (name,
                 x,
                 y,
                 digits = 3,
                 scientific = FALSE,
                 range = c("IQR","95%CI"),
                 logchange = FALSE,
                 pos=2,
                 method=c("non-parametric","parametric"),
                 total.column=FALSE, ...)
```

Arguments

name	the name of the feature.
x	the information to summarize.
y	the classification of the cohort.
digits	how many significant digits are to be used.
scientific	either a logical specifying whether result should be encoded in scientific format.
range	the range to be visualized.
logchange	either a logical specifying whether log2 of fold change should be visualized.
pos	a value indicating the position of range to be visualized. 1 for column, 2 for row.

method	a character string indicating which test method is to be computed. "non-parametric" (default), or "parametric".
total.column	option to visualize the total (by default = "FALSE")
...	further arguments to be passed to or from methods.

Value

The function returns a table with the summarized information and the relative p-value. For non-parametric method, if the number of group is equal to two, the p-value is computed using the Wilcoxon rank-sum test, Kruskal-Wallis test otherwise. For parametric method, if the number of group is equal to two, the p-value is computed using the Student's t-Test, ANOVA one-way otherwise.

Author(s)

Stefano Cacciatore

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

See Also

correlation.test, categorical.test, txtsummary

Examples

```
data(clinical)

hosp=clinical[,"Hospital"]
gender=clinical[,"Gender"]
GS=clinical[,"Gleason score"]
BMI=clinical[,"BMI"]
age=clinical[,"Age"]

A=categorical.test("Gender",gender,hosp)
B=categorical.test("Gleason score",GS,hosp)

C=continuous.test("BMI",BMI,hosp,digits=2)
D=continuous.test("Age",age,hosp,digits=1)

rbind(A,B,C,D)
```

core_cpp *Maximization of Cross-Validateed Accuracy Methods*

Description

This function performs the maximization of cross-validated accuracy by an iterative process

Usage

```
core_cpp(x,
         xTdata=NULL,
         clbest,
         Tcycle=20,
         FUN=c("PLS-DA","KNN"),
         fpar=2,
         constrain=NULL,
         fix=NULL,
         shake=FALSE)
```

Arguments

x	a matrix.
xTdata	a matrix for projections. This matrix contains samples that are not used for the maximization of the cross-validated accuracy. Their classification is obtained by predicting samples on the basis of the final classification vector.
clbest	a vector to optimize.
Tcycle	number of iterative cycles that leads to the maximization of cross-validated accuracy.
FUN	classifier to be consider. Choices are "KNN" and "PLS-DA".
fpar	parameters of the classifier. If the classifier is KNN, fpar represents the number of neighbours. If the classifier is PLS-DA, fpar represents the number of components.
constrain	a vector of nrow(data) elements. Supervised constraints can be imposed by linking some samples in such a way that if one of them is changed, all other linked samples change in the same way (*i.e.*, they are forced to belong to the same class) during the maximization of the cross-validation accuracy procedure. Samples with the same identifying constrain will be forced to stay together.
fix	a vector of nrow(data) elements. The values of this vector must be TRUE or FALSE. By default all elements are FALSE. Samples with the TRUE fix value will not change the class label defined in W during the maximization of the cross-validation accuracy procedure. For more information refer to Cacciatore, *et al.* (2014).
shake	if shake = FALSE the cross-validated accuracy is computed with the class defined in W, before the maximization of the cross-validation accuracy procedure.

Value

The function returns a list with 3 items:

`clbest`	a classification vector with a maximized cross-validated accuracy.
`accbest`	the maximum cross-validated accuracy achieved.
`vect_acc`	a vector of all cross-validated accuracies obtained.
`vect_proj`	a prediction of samples in `xTdata` matrix using the vector clbest. This output is present only if `xTdata` is not `NULL`.

Author(s)

Stefano Cacciatore and Leonardo Tenori

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

See Also

`KODAMA.matrix`,`KODAMA.visualization`

Examples

```
# Here, the famous (Fisher's or Anderson's) iris data set was loaded
data(iris)
u=as.matrix(iris[,-5])
s=sample(1:150,150,TRUE)

# The maximization of the accuracy of the vector s is performed
results=core_cpp(u, clbest=s,fpar = 5)


print(as.numeric(results$clbest))
```

`correlation.test` *Continuous Information*

Description

Summarization of the continuous information.

Usage

```
correlation.test (x,
                  y,
                  method = c("pearson", "spearman","MINE"),
                  name=NA,
                  perm=100, ...)
```

Arguments

`x`	a numeric vector.
`y`	a numeric vector.
`method`	a character string indicating which correlation method is to be computed. "pearson" (default), "spearman", or "MINE".
`name`	the name of the feature.
`perm`	number of permutation needed to estimate the p-value with MINE correlation.
`...`	further arguments to be passed to or from methods.

Value

The function returns a table with the summarized information.

Author(s)

Stefano Cacciatore

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

See Also

`categorical.test`,`continuous.test`, `txtsummary`

Examples

```
data(clinical)

correlation.test(clinical[,"Age"],clinical[,"BMI"],name="correlation between Age and BMI")
```

`dinisurface` *Ulisse Dini Data Set Generator*

Description

This function creates a data set based upon data points distributed on a Ulisse Dini's surface.

Usage

```
dinisurface(N=1000)
```

Arguments

`N`	Number of data points.

Value

The function returns a three dimensional data set.

Author(s)

Stefano Cacciatore and Leonardo Tenori

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

See Also

`helicoid`, `swissroll`, `spirals`

Examples

```
require("rgl")
x=dinisurface()
open3d()
plot3d(x, col=rainbow(1000),box=FALSE,size=3)
```

floyd *Find Shortest Paths Between All Nodes in a Graph*

Description

The floyd function finds all shortest paths in a graph using Floyd's algorithm.

Usage

```
floyd(data)
```

Arguments

data	matrix or distance object

Value

floyd returns a matrix with the total lengths of the shortest path between each pair of points.

References

Floyd, Robert W
Algorithm 97: Shortest Path.
Communications of the ACM 1962; 5 (6): 345. doi:10.1145/367766.368168.

Examples

```
# build a graph with 5 nodes
x=matrix(c(0,NA,NA,NA,NA,30,0,NA,NA,NA,10,NA,0,NA,NA,NA,70,50,0,10,NA,40,20,60,0),ncol=5)
print(x)

# compute all path lengths
z=floyd(x)
print(z)
```

frequency_matching *Frequency Matching*

Description

A method to select unbalanced groupd in a cohort.

Usage

```
frequency_matching (data,label,times=5,seed=1234)
```

Arguments

data	a data.frame of data.
label	a classification of the groups.
times	The ratio between the two groups.
seed	a single number for random number generation.

Value

The function returns a list with 2 items or 4 items (if a test data set is present):

data	the data after the frequency matching.
label	the label after the frequency matching.
selection	the rows selected for the frequency matching.

Author(s)

Stefano Cacciatore

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

Examples

```
data(clinical)

hosp=clinical[,"Hospital"]
gender=clinical[,"Gender"]
GS=clinical[,"Gleason score"]
BMI=clinical[,"BMI"]
age=clinical[,"Age"]

A=categorical.test("Gender",gender,hosp)
B=categorical.test("Gleason score",GS,hosp)

C=continuous.test("BMI",BMI,hosp,digits=2)
D=continuous.test("Age",age,hosp,digits=1)

# Analysis without matching
rbind(A,B,C,D)
```

```
# The order is important. Right is more important than left in the vector
# So, Ethnicity will be more important than Age
var=c("Age","BMI","Gleason score")
t=frequency_matching(clinical[,var],clinical[,"Hospital"],times=1)

newdata=clinical[t$selection,]

hosp.new=newdata[,"Hospital"]
gender.new=newdata[,"Gender"]
GS.new=newdata[,"Gleason score"]
BMI.new=newdata[,"BMI"]
age.new=newdata[,"Age"]

A=categorical.test("Gender",gender.new,hosp.new)
B=categorical.test("Gleason score",GS.new,hosp.new)

C=continuous.test("BMI",BMI.new,hosp.new,digits=2)
D=continuous.test("Age",age.new,hosp.new,digits=1)

# Analysis with matching
rbind(A,B,C,D)
```

`helicoid` *Helicoid Data Set Generator*

Description

This function creates a data set based upon data points distributed on a Helicoid surface.

Usage

```
helicoid(N=1000)
```

Arguments

`N`	Number of data points.

Value

The function returns a three dimensional data set.

Author(s)

Stefano Cacciatore and Leonardo Tenori

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

See Also

swissroll,dinisurface,spirals

Examples

```
require("rgl")
x=helicoid()
open3d()
plot3d(x, col=rainbow(1000),box=FALSE,size=3)
```

---

k.test *K-Test of Statistical Association*

Description

This function performs a permutation test using PLS to assess association between the KODAMA output and any additional related parameters such as clinical metadata.

Usage

```
k.test(data, labels, n = 100)
```

Arguments

data	a matrix.
labels	a classification vector.
n	number of iterations of the permutation test.

Value

The p-value of the test.

Author(s)

Stefano Cacciatore

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

See Also

KODAMA.matrix,KODAMA.visualization

Examples

```
 data(iris)
data=iris[,-5]
labels=iris[,5]
kk=KODAMA.matrix(data,FUN="KNN",f.par=2)
kkplot=KODAMA.visualization(kk,"t-SNE")
k1=k.test(kkplot,labels)
print(k1)
k2=k.test(kkplot,sample(labels))
print(k2)
```

`knn.double.cv` *Cross-Validation with k-Nearest Neighbors algorithm.*

Description

This function performs a 10-fold cross validation on a given data set using *k*-Nearest Neighbors (*k*NN) model. To assess the prediction ability of the model, a 10-fold cross-validation is conducted by generating splits with a ratio 1:9 of the data set, that is by removing 10% of samples prior to any step of the statistical analysis, including PLS component selection and scaling. Best number of component for PLS was carried out by means of 10-fold cross-validation on the remaining 90% selecting the best Q2y value. Permutation testing was undertaken to estimate the classification/regression performance of predictors.

Usage

```
knn.double.cv(Xdata,
              Ydata,
              constrain=1:nrow(Xdata),
              compmax=min(5,c(ncol(Xdata),nrow(Xdata))),
              perm.test=FALSE,
              optim=TRUE,
              scaling = c("centering","autoscaling"),
```

```
                times=100,
                runn=10)
```

Arguments

Xdata	a matrix.
Ydata	the responses. If Ydata is a numeric vector, a regression analysis will be performed. If Ydata is factor, a classification analysis will be performed.
constrain	a vector of nrow(data) elements. Sample with the same identifying constrain will be split in the training set or in the test set of cross-validation together.
compmax	the number of k to be used for classification.
perm.test	a classification vector.
optim	if perform the optmization of the number of k.
scaling	the scaling method to be used. Choices are "centering" or "autoscaling" (by default = "centering"). A partial string sufficient to uniquely identify the choice is permitted.
times	number of cross-validations with permutated samples
runn	number of cross-validations loops.

Value

A list with the following components:

Ypred	the vector containing the predicted values of the response variables obtained by cross-validation.
Yfit	the vector containing the fitted values of the response variables.
Q2Y	Q2y value.
R2Y	R2y value.
conf	The confusion matrix (only in classification mode).
acc	The cross-validated accuracy (only in classification mode).
txtQ2Y	a summary of the Q2y values.
txtR2Y	a summary of the R2y values.

Author(s)

Stefano Cacciatore

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

Examples

```
data(iris)
data=iris[,-5]
labels=iris[,5]
pp=knn.double.cv(data,labels)
print(pp$Q2Y)
table(pp$Ypred,labels)


data(MetRef)
u=MetRef$data;
u=u[,-which(colSums(u)==0)]
u=normalization(u)$newXtrain
u=scaling(u)$newXtrain
pp=knn.double.cv(u,as.factor(MetRef$donor))
print(pp$Q2Y)
table(pp$Ypred,MetRef$donor)
```

---

knn.kodama *k-Nearest Neighbors Classifier.*

Description

k-nearest neighbour classification for a test set from a training set.

Usage

```
knn.kodama(Xtrain,
           Ytrain,
           Xtest,
           Ytest=NULL,
           k,
           scaling = c("centering","autoscaling"),
           perm.test=FALSE,
           times=1000)
```

Arguments

Xtrain	a matrix of training set cases.
Ytrain	a classification vector.
Xtest	a matrix of test set cases.
Ytest	a classification vector.
k	the number of nearest neighbors to consider.

scaling	the scaling method to be used. Choices are "centering" or "autoscaling" (by default = "centering"). A partial string sufficient to uniquely identify the choice is permitted.
perm.test	a classification vector.
times	a classification vector.

Details

The function utilizes the Approximate Nearest Neighbor (ANN) C++ library, which can give the exact nearest neighbours or (as the name suggests) approximate nearest neighbours to within a specified error bound. For more information on the ANN library please visit http://www.cs.umd.edu/~mount/ANN/.

Value

The function returns a vector of predicted labels.

Author(s)

Stefano Cacciatore and Leonardo Tenori

References

Bentley JL (1975)
Multidimensional binary search trees used for associative search.
Communication ACM 1975;18:309-517.

Arya S, Mount DM
Approximate nearest neighbor searching
Proc. 4th Ann. ACM-SIAM Symposium on Discrete Algorithms (SODA'93);271-280.

Arya S, Mount DM, Netanyahu NS, Silverman R, Wu AY
An optimal algorithm for approximate nearest neighbor searching
Journal of the ACM 1998;45:891-923.

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

See Also

KODAMA.matrix,KODAMA.visualization

Examples

```
data(iris)
data=iris[,-5]
labels=iris[,5]
ss=sample(150,15)

z=knn.kodama(data[-ss,], labels[-ss], data[ss,], k=5)
table(z$Ypred[,5],labels[ss])
```

KODAMA.matrix *Knowledge Discovery by Accuracy Maximization*

Description

KODAMA (KnOwledge Discovery by Accuracy MAximization) is an unsupervised and semi-supervised learning algorithm that performs feature extraction from noisy and high-dimensional data.

Usage

```
KODAMA.matrix (data,
               M = 100,
               Tcycle = 20,
               FUN_VAR = function(x) {  ceiling(ncol(x)) },
               FUN_SAM = function(x) {  ceiling(nrow(x) * 0.75)},
               bagging = FALSE,
               FUN = c("PLS-DA","KNN"),
               f.par = 5,
               W = NULL,
               constrain = NULL,
               fix=NULL,
               epsilon = 0.05,
               dims=2,
               landmarks=1000,
               neighbors=min(c(landmarks,nrow(data)))-1)
```

Arguments

data	a matrix.
M	number of iterative processes (step I-III).
Tcycle	number of iterative cycles that leads to the maximization of cross-validated accuracy.
FUN_VAR	function to select the number of variables to select randomly. By default all variable are taken.
FUN_SAM	function to select the number of samples to select randomly. By default the 75 per cent of all samples are taken.

bagging	Should sampling be with replacement, bagging = TRUE. By default bagging = FALSE.
FUN	classifier to be considered. Choices are "PLS-DA" and "KNN".
f.par	parameters of the classifier.
W	a vector of nrow(data) elements. The KODAMA procedure can be started by different initializations of the vector W. Without any *a priori* information the vector W can be initialized with each element being different from the others (*i.e.*, each sample categorized in a one-element class). Alternatively, the vector W can be initialized by a clustering procedure, such as kmeans.
constrain	a vector of nrow(data) elements. Supervised constraints can be imposed by linking some samples in such a way that if one of them is changed the remaining linked samples must change in the same way (*i.e.*, they are forced to belong to the same class) during the maximization of the cross-validation accuracy procedure. Samples with the same identifying constrain will be forced to stay together.
fix	a vector of nrow(data) elements. The values of this vector must to be TRUE or FALSE. By default all elements are FALSE. Samples with the TRUE fix value will not change the class label defined in W during the maximization of the cross-validation accuracy procedure.
epsilon	cut-off value for low proximity. High proximity are typical of intracluster relationships, whereas low proximities are expected for intercluster relationships. Very low proximities between samples are ignored by (default) setting epsilon = 0.05.
dims	dimensions of the configurations of t-SNE based on the KODAMA dissimilarity matrix.
landmarks	number of landmarks to use.
neighbors	number of neighbors to include in the dissimilarity matrix yo pass to the KODAMA.visualization function.

Details

KODAMA consists of five steps. These can be in turn divided into two parts: (i) the maximization of cross-validated accuracy by an iterative process (step I and II), resulting in the construction of a proximity matrix (step III), and (ii) the definition of a dissimilarity matrix (step IV and V). The first part entails the core idea of KODAMA, that is, the partitioning of data guided by the maximization of the cross-validated accuracy. At the beginning of this part, a fraction of the total samples (defined by FUN_SAM) are randomly selected from the original data. The whole iterative process (step I-III) is repeated M times to average the effects owing to the randomness of the iterative procedure. Each time that this part is repeated, a different fraction of samples is selected. The second part aims at collecting and processing these results by constructing a dissimilarity matrix to provide a holistic view of the data while maintaining their intrinsic structure (steps IV and V). Then, KODAMA.visualization function is used to visualise the results of KODAMA dissimilarity matrix.

Value

The function returns a list with 4 items:

dissimilarity	a dissimilarity matrix.
acc	a vector with the M cross-validated accuracies.
proximity	a proximity matrix.
v	a matrix containing the all classification obtained maximizing the cross-validation accuracy.
res	a matrix containing all classification vectors obtained through maximizing the cross-validation accuracy.
f.par	parameters of the classifier..
entropy	Shannon's entropy of the KODAMA proximity matrix.
landpoints	indexes of the landmarks used.
data	original data.
knn_Armadillo	dissimilarity matrix used as input for the KODAMA.visualization function.

Author(s)

Stefano Cacciatore and Leonardo Tenori

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

L.J.P. van der Maaten and G.E. Hinton.
Visualizing High-Dimensional Data Using t-SNE.
Journal of Machine Learning Research 9 (Nov) : 2579-2605, 2008.

L.J.P. van der Maaten.
Learning a Parametric Embedding by Preserving Local Structure.
In Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS), JMLR W&CP 5:384-391, 2009.

McInnes L, Healy J, Melville J.
Umap: Uniform manifold approximation and projection for dimension reduction.
arXiv preprint:1802.03426. 2018 Feb 9.

See Also

KODAMA.visualization

Examples

```
data(iris)
data=iris[,-5]
labels=iris[,5]
kk=KODAMA.matrix(data,FUN="KNN",f.par=2)
cc=KODAMA.visualization(kk,"t-SNE")
plot(cc,col=as.numeric(labels),cex=2)
```

KODAMA.visualization *Visualization of KODAMA output*

Description

Provides a simple function to transform the KODAMA dissimilarity matrix in a low-dimensional space.

Usage

```
KODAMA.visualization(kk,
                     method=c("t-SNE","MDS","UMAP"),
                   perplexity=min(30,floor((kk$knn_Armadillo$neighbors+1)/3)-1),
                     ...)
```

Arguments

kk	output of KODAMA.matrix function.
method	method to be considered for transforming the dissimilarity matrix in a low-dimensional space. Choices are "t-SNE", "MDS", and "UMAP".
perplexity	Perplexity parameter. (optimal number of neighbors) for "t-SNE" only.
...	other parameters for "t-SNE", "MDS", or "UMAP".

Value

The function returns a matrix contains the coordinates of the datapoints in a low-dimensional space.

Author(s)

Stefano Cacciatore and Leonardo Tenori

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

L.J.P. van der Maaten and G.E. Hinton.
Visualizing High-Dimensional Data Using t-SNE.
Journal of Machine Learning Research 9 (Nov) : 2579-2605, 2008.

L.J.P. van der Maaten.
Learning a Parametric Embedding by Preserving Local Structure.
In Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS), JMLR W&CP 5:384-391, 2009.

McInnes L, Healy J, Melville J.
Umap: Uniform manifold approximation and projection for dimension reduction.
arXiv preprint:1802.03426. 2018 Feb 9.

See Also

KODAMA.visualization

Examples

```
data(iris)
data=iris[,-5]
labels=iris[,5]
kk=KODAMA.matrix(data,FUN="KNN",f.par=2)
cc=KODAMA.visualization(kk,"t-SNE")
plot(cc,col=as.numeric(labels),cex=2)
```

---

loads *Variable Ranking*

Description

This function can be used to extract the variable ranking when KODAMA is performed with the PLS-DA classifier.

Usage

```
loads(model,method=c("loadings","kruskal.test"))
```

Arguments

model	output of KODAMA.
method	method to be used. Choices are "loadings" and "kruskal.test".

Value

The function returns a vector of values indicating the "importance" of each variable. If "method="loadings" the average of the loading of the first component of PLS models based on the cross-validated accuracy maximized vector is computed. If "method="kruskal.test" the average of minus logarithm of p-value of Kruskal-Wallis Rank Sum test is computed.

Author(s)

Stefano Cacciatore

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

See Also

KODAMA.matrix,KODAMA.visualization

Examples

```
data(iris)
data=iris[,-5]
labels=iris[,5]
kk=KODAMA.matrix(data,FUN="PLS-DA")
loads(kk)
```

`lymphoma` *Lymphoma Gene Expression Dataset*

Description

This dataset consists of gene expression profiles of the three most prevalent adult lymphoid malignancies: diffuse large B-cell lymphoma (DLBCL), follicular lymphoma (FL), and B-cell chronic lymphocytic leukemia (B-CLL). The dataset consists of 4,682 mRNA genes for 62 samples (42 samples of DLBCL, 9 samples of FL, and 11 samples of B-CLL). Missing value are imputed and data are standardized as described in Dudoit, *et al*. (2002).

Usage

```
data(lymphoma)
```

Value

A list with the following elements:

`data`	Gene expression data. A matrix with 62 rows and 4,682 columns.
`class`	Class index. A vector with 62 elements.

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

Alizadeh AA, Eisen MB, Davis RE, *et al.*
Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling.
Nature 2000;403(6769):503-511.

Dudoit S, Fridlyand J, Speed TP
Comparison of discrimination methods for the classification of tumors using gene expression data.
J Am Stat Assoc 2002;97(417):77-87.

Examples

```
data(lymphoma)
class=1+as.numeric(lymphoma$class)
cc=pca(lymphoma$data)$x
plot(cc,pch=21,bg=class)

kk=KODAMA.matrix(lymphoma$data)
```

```
cc=KODAMA.visualization(kk,"t-SNE")

plot(cc,pch=21,bg=class)
```

`mcplot` *Evaluation of the Monte Carlo accuracy results*

Description

This function can be used to plot the accuracy values obtained during KODAMA procedure.

Usage

```
mcplot(model)
```

Arguments

`model`	output of KODAMA.

Value

No return value.

Author(s)

Stefano Cacciatore

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

See Also

`KODAMA.matrix`,`KODAMA.visualization`

Examples

```
data=as.matrix(iris[,-5])
kk=KODAMA.matrix(data)
mcplot(kk)
```

MetRef *Nuclear Magnetic Resonance Spectra of Urine Samples*

Description

The data belong to a cohort of 22 healthy donors (11 male and 11 female) where each provided about 40 urine samples over the time course of approximately 2 months, for a total of 873 samples. Each sample was analysed by Nuclear Magnetic Resonance Spectroscopy. Each spectrum was divided in 450 spectral bins.

Usage

```
data(MetRef)
```

Value

A list with the following elements:

`data`	Metabolomic data. A matrix with 873 rows and 450 columns.
`gender`	Gender index. A vector with 873 elements.
`donor`	Donor index. A vector with 873 elements.

References

Assfalg M, Bertini I, Colangiuli D, *et al.*
Evidence of different metabolic phenotypes in humans.
Proc Natl Acad Sci U S A 2008;105(5):1420-4. doi: 10.1073/pnas.0705685105. Link

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

Examples

```
data(MetRef)
u=MetRef$data;
u=u[,-which(colSums(u)==0)]
u=normalization(u)$newXtrain
u=scaling(u)$newXtrain
class=as.numeric(as.factor(MetRef$gender))
cc= pca(u)$x
plot(cc,pch=21,bg=class)

class=as.numeric(as.factor(MetRef$donor))
```

```
plot(cc,pch=21,bg=rainbow(22)[class])

kk=KODAMA.matrix(u)
cc=KODAMA.visualization(kk,"t-SNE")
plot(cc,pch=21,bg=rainbow(22)[class])
```

`multi_analysis` *Continuous Information*

Description

Summarization of the continuous information.

Usage

```
multi_analysis  (data,
                 y,
                 FUN=c("continuous.test","correlation.test"), ...)
```

Arguments

`data`	the matrix containing the continuous values. Each row corresponds to a different sample. Each column corresponds to a different variable.
`y`	the classification of the cohort.
`FUN`	function to be considered. Choices are "`continuous.test`" and "`correlation.test`"
`...`	further arguments to be passed to or from methods.

Value

The function returns a table with the summarized information. If the number of group is equal to two, the p-value is computed using the Wilcoxon rank-sum test, Kruskal-Wallis test otherwise.

Author(s)

Stefano Cacciatore

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

See Also

categorical.test,continuous.test,correlation.test, txtsummary

Examples

```
data(clinical)


multi_analysis(clinical[,c("BMI","Age")],clinical[,"Hospital"],FUN="continuous.test")
```

---

normalization *Normalization Methods*

Description

Collection of Different Normalization Methods.

Usage

```
normalization(Xtrain,Xtest=NULL, method = "pqn",ref=NULL)
```

Arguments

Xtrain	a matrix of data (training data set).
Xtest	a matrix of data (test data set).(by default = NULL).
method	the normalization method to be used. Choices are "none", "pqn", "sum", "median", "sqrt" (by default = "pqn"). A partial string sufficient to uniquely identify the choice is permitted.
ref	Reference sample for Probabilistic Quotient Normalization. (by default = NULL).

Details

A number of different normalization methods are provided:

- "none": no normalization method is applied.
- "pqn": the Probabilistic Quotient Normalization is computed as described in *Dieterle, et al.* (2006).
- "sum": samples are normalized to the sum of the absolute value of all variables for a given sample.
- "median": samples are normalized to the median value of all variables for a given sample.
- "sqrt": samples are normalized to the root of the sum of the squared value of all variables for a given sample.

Value

The function returns a list with 2 items or 4 items (if a test data set is present):

`newXtrain`	a normalized matrix (training data set).
`coeXtrain`	a vector of normalization coefficient of the training data set.
`newXtest`	a normalized matrix (test data set).
`coeXtest`	a vector of normalization coefficient of the test data set.

Author(s)

Stefano Cacciatore and Leonardo Tenori

References

Dieterle F,Ross A, Schlotterbeck G, Senn H.
Probabilistic Quotient Normalization as Robust Method to Account for Diluition of Complex Biological Mixtures. Application in 1H NMR Metabolomics.
Anal Chem 2006;78:4281-90.

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

See Also

`scaling`

Examples

```
data(MetRef)
u=MetRef$data;
u=u[,-which(colSums(u)==0)]
u=normalization(u)$newXtrain
u=scaling(u)$newXtrain
class=as.numeric(as.factor(MetRef$gender))
cc=pca(u)
plot(cc$x,pch=21,bg=class)
```

pca *Principal Components Analysis*

Description

Performs a principal components analysis on the given data matrix and returns the results as an object of class "prcomp".

Usage

```
pca(x, ...)
```

Arguments

x	a matrix of data.
...	arguments passed to prcomp function.

Value

The function returns a list with class prcomp containing the following components:

sdev	the standard deviations of the principal components (i.e., the square roots of the eigenvalues of the covariance/correlation matrix, though the calculation is actually done with the singular values of the data matrix).
rotation	the matrix of variable loadings (i.e., a matrix whose columns contain the eigenvectors). The function princomp returns this in the element loadings.
x	if retx is TRUE the value of the rotated data (the centred (and scaled if requested) data multiplied by the rotation matrix) is returned. Hence, cov(x) is the diagonal matrix diag(sdev^2). For the formula method, napredict() is applied to handle the treatment of values omitted by the na.action.
center, scale	the centering and scaling used, or FALSE.
txt	the component of variance of each Principal Component.

Author(s)

Stefano Cacciatore

References

Pearson, K
On Lines and Planes of Closest Fit to Systems of Points in Space.
Philosophical Magazine 1901;2 (11): 559-572. doi:10.1080/14786440109462720. Link

See Also

prcomp

Examples

```
data(MetRef)
u=MetRef$data;
u=u[,-which(colSums(u)==0)]
u=normalization(u)$newXtrain
u=scaling(u)$newXtrain
class=as.numeric(as.factor(MetRef$gender))
cc=pca(u)
plot(cc$x,pch=21,bg=class)
```

---

pls.double.cv *Cross-Validation with PLS-DA.*

Description

This function performs a 10-fold cross validation on a given data set using Partial Least Squares (PLS) model. To assess the prediction ability of the model, a 10-fold cross-validation is conducted by generating splits with a ratio 1:9 of the data set, that is by removing 10% of samples prior to any step of the statistical analysis, including PLS component selection and scaling. Best number of component for PLS was carried out by means of 10-fold cross-validation on the remaining 90% selecting the best Q2y value. Permutation testing was undertaken to estimate the classification/regression performance of predictors.

Usage

```
pls.double.cv(Xdata,
              Ydata,
              constrain=1:nrow(Xdata),
              compmax=min(5,c(ncol(Xdata),nrow(Xdata))),
              perm.test=FALSE,
              optim=TRUE,
              scaling = c("centering","autoscaling"),
              times=100,
              runn=10)
```

Arguments

Xdata	a matrix.
Ydata	the responses. If Ydata is a numeric vector, a regression analysis will be performed. If Ydata is factor, a classification analysis will be performed.
constrain	a vector of nrow(data) elements. Sample with the same identifying constrain will be split in the training set or in the test set of cross-validation together.
compmax	the number of latent components to be used for classification.
perm.test	a classification vector.
optim	if perform the optmization of the number of components.

scaling	the scaling method to be used. Choices are "centering" or "autoscaling" (by default = "centering"). A partial string sufficient to uniquely identify the choice is permitted.
times	number of cross-validations with permutated samples
runn	number of cross-validations loops.

Value

A list with the following components:

B	the (p x m x length(ncomp)) array containing the regression coefficients. Each row corresponds to a predictor variable and each column to a response variable. The third dimension of the matrix B corresponds to the number of PLS components used to compute the regression coefficients. If ncomp has length 1, B is just a (p x m) matrix.
Ypred	the vector containing the predicted values of the response variables obtained by cross-validation.
Yfit	the vector containing the fitted values of the response variables.
P	the (p x max(ncomp)) matrix containing the X-loadings.
Q	the (m x max(ncomp)) matrix containing the Y-loadings.
T	the (ntrain x max(ncomp)) matrix containing the X-scores (latent components)
R	the (p x max(ncomp)) matrix containing the weights used to construct the latent components.
Q2Y	Q2y value.
R2Y	R2y value.
R2X	vector containg the explained variance of X by each PLS component.
txtQ2Y	a summary of the Q2y values.
txtR2Y	a summary of the R2y values.

Author(s)

Stefano Cacciatore

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

Examples

```
data(iris)
data=iris[,-5]
labels=iris[,5]
pp=pls.double.cv(data,labels)
print(pp$Q2Y)
table(pp$Ypred,labels)

data(MetRef)
u=MetRef$data;
u=u[,-which(colSums(u)==0)]
u=normalization(u)$newXtrain
u=scaling(u)$newXtrain
pp=pls.double.cv(u,as.factor(MetRef$donor))
print(pp$Q2Y)
table(pp$Ypred,MetRef$donor)
```

`pls.kodama` *Partial Least Squares regression.*

Description

Partial Least Squares (PLS) regression for test set from training set.

Usage

```
pls.kodama(Xtrain,
           Ytrain,
           Xtest,
           Ytest = NULL,
           ncomp,
           scaling = c("centering","autoscaling"),
           perm.test=FALSE,
           times=1000)
```

Arguments

`Xtrain`	a matrix of training set cases.
`Ytrain`	a classification vector.
`Xtest`	a matrix of test set cases.
`Ytest`	a classification vector.
`ncomp`	the number of components to consider.

scaling	the scaling method to be used. Choices are "centering" or "autoscaling" (by default = "centering"). A partial string sufficient to uniquely identify the choice is permitted.
perm.test	a classification vector.
times	a classification vector.

Value

A list with the following components:

B	the (p x m x length(ncomp)) matrix containing the regression coefficients. Each row corresponds to a predictor variable and each column to a response variable. The third dimension of the matrix B corresponds to the number of PLS components used to compute the regression coefficients. If ncomp has length 1, B is just a (p x m) matrix.
Ypred	the (ntest x m x length(ncomp)) containing the predicted values of the response variables for the observations from Xtest. The third dimension of the matrix Ypred corresponds to the number of PLS components used to compute the regression coefficients.
P	the (p x max(ncomp)) matrix containing the X-loadings.
Q	the (m x max(ncomp)) matrix containing the Y-loadings.
T	the (ntrain x max(ncomp)) matrix containing the X-scores (latent components)
R	the (p x max(ncomp)) matrix containing the weights used to construct the latent components.

Author(s)

Stefano Cacciatore

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

See Also

KODAMA.matrix,KODAMA.visualization

Examples

```
data(iris)
data=iris[,-5]
labels=iris[,5]
ss=sample(150,15)
ncomponent=3

z=pls.kodama(data[-ss,], labels[-ss], data[ss,], ncomp=ncomponent)
table(z$Ypred[,ncomponent],labels[ss])
```

`scaling` *Scaling Methods*

Description

Collection of Different Scaling Methods.

Usage

```
scaling(Xtrain,Xtest=NULL, method = "autoscaling")
```

Arguments

`Xtrain`	a matrix of data (training data set).
`Xtest`	a matrix of data (test data set).(by default = NULL).
`method`	the scaling method to be used. Choices are "`none`", "`centering`", "`autoscaling`", "`rangescaling`", "`paretoscaling`" (by default = "`autoscaling`"). A partial string sufficient to uniquely identify the choice is permitted.

Details

A number of different scaling methods are provided:

- "`none`": no scaling method is applied.
- "`centering`": centers the mean to zero.
- "`autoscaling`": centers the mean to zero and scales data by dividing each variable by the variance.
- "`rangescaling`": centers the mean to zero and scales data by dividing each variable by the difference between the minimum and the maximum value.
- "`paretoscaling`": centers the mean to zero and scales data by dividing each variable by the square root of the standard deviation. Unit scaling divides each variable by the standard deviation so that each variance equal to 1.

Value

The function returns a list with 1 item or 2 items (if a test data set is present):

`newXtrain`	a scaled matrix (training data set).
`newXtest`	a scale matrix (test data set).

Author(s)

Stefano Cacciatore and Leonardo Tenori

References

van den Berg RA, Hoefsloot HCJ, Westerhuis JA, *et al.*
Centering, scaling, and transformations: improving the biological information content of metabolomics data.
BMC Genomics 2006;7(1):142.

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

See Also

`normalization`

Examples

```
data(MetRef)
u=MetRef$data;
u=u[,-which(colSums(u)==0)]
u=normalization(u)$newXtrain
u=scaling(u)$newXtrain
class=as.numeric(as.factor(MetRef$gender))
cc=pca(u)
plot(cc$x,pch=21,bg=class,xlab=cc$txt[1],ylab=cc$txt[2])
```

---

spirals *Spirals Data Set Generator*

Description

Produces a data set of spiral clusters.

Usage

```
spirals(n=c(100,100,100),sd=c(0,0,0))
```

Arguments

n	a vector of integer. The length of the vector is the number of clusters and each number corresponds to the number of data points in each cluster.
sd	amount of noise for each spiral.

Value

The function returns a two dimensional data set.

Author(s)

Stefano Cacciatore and Leonardo Tenori

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

See Also

helicoid,dinisurface,swissroll

Examples

```
v1=spirals(c(100,100,100),c(0.1,0.1,0.1))
plot(v1,col=rep(2:4,each=100))
v2=spirals(c(100,100,100),c(0.1,0.2,0.3))
plot(v2,col=rep(2:4,each=100))
v3=spirals(c(100,100,100,100,100),c(0,0,0.2,0,0))
plot(v3,col=rep(2:6,each=100))
v4=spirals(c(20,40,60,80,100),c(0.1,0.1,0.1,0.1,0.1))
plot(v4,col=rep(2:6,c(20,40,60,80,100)))
```

`swissroll` *Swiss Roll Data Set Generator*

Description

Computes the Swiss Roll data set of a given number of data points.

Usage

```
swissroll(N=1000)
```

Arguments

`N`	Number of data points.

Value

The function returns a three dimensional matrix.

Author(s)

Stefano Cacciatore and Leonardo Tenori

References

Balasubramanian M, Schwartz EL
The isomap algorithm and topological stability.
Science 2002;295(5552):7.

Roweis ST, Saul LK
Nonlinear dimensionality reduction by locally linear embedding.
Science 2000;290(5500):2323-6.

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

See Also

`helicoid`,`dinisurface`,`spirals`

Examples

```
require("rgl")
x=swissroll()
open3d()
plot3d(x, col=rainbow(1000),box=FALSE,size=3)
```

---

transformy *Conversion Classification Vector to Matrix*

Description

This function converts a classification vector into a classification matrix.

Usage

```
transformy(y)
```

Arguments

y	a vector or factor.

Details

This function converts a classification vector into a classification matrix.

Value

A matrix.

Author(s)

Stefano Cacciatore and Leonardo Tenori

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

Examples

```
y=rep(1:10,3)
print(y)
z=transformy(y)
print(z)
```

`txtsummary` *Median and Coefficient Interval*

Description

Summarization of a numeric vector.

Usage

```
txtsummary (x,digits=0,scientific=FALSE,range=c("IQR","95%CI"))
```

Arguments

`x`	a numeric vector.
`digits`	how many significant digits are to be used.
`scientific`	either a logical specifying whether result should be encoded in scientific format.
`range`	the range to be visualized.

Value

The function returns the median and the range (interquartile or 95% coefficient interval) of numeric vetor.

Author(s)

Stefano Cacciatore

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

See Also

`categorical.test`,`continuous.test`,`correlation.test`, `txtsummary`

Examples

```
data(clinical)

txtsummary(clinical[,"BMI"])
```

USA *State of the Union Data Set*

Description

This dataset consists of the spoken, not written, addresses from 1900 until the sixth address by Barack Obama in 2014. Punctuation characters, numbers, words shorter than three characters, and stop-words (e.g., "that", "and", and "which") were removed from the dataset. This resulted in a dataset of 86 speeches containing 834 different meaningful words each. Term frequency-inverse document frequency (TF-IDF) was used to obtain feature vectors. It is often used as a weighting factor in information retrieval and text mining. The TF-IDF value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to control for the fact that some words are generally more common than others.

Usage

```
data(USA)
```

Value

A list with the following elements:

`data`	TF-IDF data. A matrix with 86 rows and 834 columns.
`year`	Year index. A vector with 86 elements.
`president`	President index. A vector with 86 elements.

Author(s)

Stefano Cacciatore and Leonardo Tenori

References

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-22. doi: 10.1073/pnas.1220873111. Link

Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA
KODAMA: an updated R package for knowledge discovery and data mining.
Bioinformatics 2017;33(4):621-623. doi: 10.1093/bioinformatics/btw705. Link

Examples

```
# Here is reported the analysis on the State of the Union
# of USA president as shown in Cacciatore, et al. (2014)

data(USA)
kk=KODAMA.matrix(USA$data,FUN="KNN")
cc=KODAMA.visualization(kk,"t-SNE",perplexity = 10)
```

```
oldpar <- par(cex=0.5,mar=c(15,6,2,2));
plot(USA$year,cc[,1],axes=FALSE,pch=20,xlab="",ylab="First Component");
axis(1,at=USA$year,labels=rownames(USA$data),las=2);
axis(2,las=2);
box()

par(oldpar)
```

Index