Package: KODAMA (via r-universe)

October 27, 2026

Version 3.3

Date 2026-03-17

Maintainer Stefano Cacciatore <tkcaccia@gmail.com>

Title Knowledge Discovery by Accuracy Maximization

Description A self-guided, weakly supervised learning algorithm for feature extraction from noisy and high-dimensional data. It facilitates the identification of patterns that reflect underlying group structures across all samples in a dataset. The method incorporates a novel strategy to integrate spatial information, improving the clarity of results in spatially resolved data.

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

Imports Rcpp (>= 0.12.4), Rnanoflann, methods, Matrix

LinkingTo Rcpp, RcppArmadillo, Rnanoflann, Matrix

Suggests rgl, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

SuggestsNote No suggestions

LazyData true

LazyDataCompression xz

Config/testthat/edition 3

License GPL (>= 2)

NeedsCompilation yes

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

Repository https://cran.r-universe.dev

Date/Publication 2026-03-17 15:30:02 UTC

RemoteUrl https://github.com/cran/KODAMA

RemoteRef HEAD

RemoteSha 66ba26a3fc6de2d15fb4c1f4e72e7e7567ecb684

Contents

config.tsne.default . . . 2
config.umap.default . . . 3
core_cpp . . . 5
dinisurface . . . 7
floyd . . . 8
helicoid . . . 9
kabsch . . . 10
KODAMA.matrix . . . 11
KODAMA.visualization . . . 13
lymphoma . . . 15
mcplot . . . 16
MDS.defaults . . . 17
MetRef . . . 18
normalization . . . 19
pca . . . 21
scaling . . . 22
spirals . . . 23
swissroll . . . 24
transformy . . . 25
USA . . . 26

Index **28**

`config.tsne.default` *Default configuration for Rtsne*

Description

A list with parameters customizing a Rtsne embedding. Each component of the list is an effective argument for Rtsne_neighbors().

Usage

```
config.tsne.default
```

Format

An object of class `config.tsne.default` of length 12.

Details

dims: integer, Output dimensionality

perplexity: numeric, Perplexity parameter (should not be bigger than 3 * perplexity < nrow(X) - 1, see details for interpretation)

theta: numeric, Speed/accuracy trade-off (increase for less accuracy), set to 0.0 for exact TSNE

max_iter: integer, Number of iterations

verbose: logical, Whether progress updates should be printed (default: global "verbose" option, or FALSE if that is not set)

Y_init: matrix, Initial locations of the objects. If NULL, random initialization will be used (default: NULL). Note that when using this, the initial stage with exaggerated perplexity values and a larger momentum term will be skipped.

momentum: numeric, Momentum used in the first part of the optimization

final_momentum: numeric, Momentum used in the final part of the optimization

eta: numeric, Learning rate

exaggeration_factor:

num_threads: integer, Number of threads to use when using OpenMP, default is 1. Setting to 0 corresponds to detecting and using all available cores

define.n.cores: logical. If FALSE (default), `KODAMA.visualization` overrides `num_threads` using `kk$n.cores` from `KODAMA.matrix` output.

Examples

```
# display all default settings
config.tsne.default

# create a new settings object with perplexity set to 100
custom.settings = config.tsne.default
custom.settings$perplexity = 100
custom.settings
```

`config.umap.default` *Default configuration for umap*

Description

A list with parameters customizing a Rumap embedding. Each component of the list is an effective argument for Rumap_neighbors().

Usage

```
config.umap.default
```

Format

An object of class `config.umap.default` of length 25.

Details

n_neighbors: integer; number of nearest neighbors

n_components: integer; dimension of target (output) space

metric: character or function; determines how distances between data points are computed. When using a string, available metrics are: euclidean, manhattan. Other available generalized metrics are: cosine, pearson, pearson2. Note the triangle inequality may not be satisfied by some generalized metrics, hence knn search may not be optimal. When using metric.function as a function, the signature must be function(matrix, origin, target) and should compute a distance between the origin column and the target columns

n_epochs: integer; number of iterations performed during layout optimization

input: character, use either "data" or "dist"; determines whether the primary input argument to umap() is treated as a data matrix or as a distance matrix

init: character or matrix. The default string "spectral" computes an initial embedding using eigenvectors of the connectivity graph matrix. An alternative is the string "random", which creates an initial layout based on random coordinates. This setting.can also be set to a matrix, in which case layout optimization begins from the provided coordinates.

min_dist: numeric; determines how close points appear in the final layout

set_op_ratio_mix_ratio: numeric in range [0,1]; determines who the knn-graph is used to create a fuzzy simplicial graph

local_connectivity: numeric; used during construction of fuzzy simplicial set

bandwidth: numeric; used during construction of fuzzy simplicial set

alpha: numeric; initial value of "learning rate" of layout optimization

gamma: numeric; determines, together with alpha, the learning rate of layout optimization

negative_sample_rate: integer; determines how many non-neighbor points are used per point and per iteration during layout optimization

a: numeric; contributes to gradient calculations during layout optimization. When left at NA, a suitable value will be estimated automatically.

b: numeric; contributes to gradient calculations during layout optimization. When left at NA, a suitable value will be estimated automatically.

spread: numeric; used during automatic estimation of a/b parameters.

random_state: integer; seed for random number generation used during umap()

transform_state: integer; seed for random number generation used during predict()

knn: object of class umap.knn; precomputed nearest neighbors

knn.repeat: number of times to restart knn search

verbose: logical or integer; determines whether to show progress messages

umap_learn_args: vector of arguments to python package umap-learn

define.n.cores: logical. If FALSE (default), `KODAMA.visualization` overrides `n_threads` and `n_sgd_threads` using `kk$n.cores` from `KODAMA.matrix` output.

Examples

```
# display all default settings
config.umap.default

# create a new settings object with n_neighbors set to 5
custom.settings = config.umap.default
custom.settings$n_neighbors = 5
custom.settings
```

---

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,
         f.par.pls = 5,
         constrain = NULL,
         fix = NULL,
         ...)
```

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.
f.par.pls	Number of PLS components. The backend is selected automatically inside corecpp before each cross-validation step: plssvd when f.par.pls is smaller than the current number of classes, otherwise simpls.
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).
...	Ignored legacy arguments. Passing FUN is deprecated and has no effect.

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

Abdel-Shafy EA, Kassim M, Vignol A, *et al.*
KODAMA enables self-guided weakly supervised learning in spatial transcriptomics.
bioRxiv 2025. doi: 10.1101/2025.05.28.656544. doi:10.1101/2025.05.28.656544

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-5122. doi: 10.1073/pnas.1220873111. doi:10.1073/pnas.1220873111

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. doi:10.1093/bioinformatics/btw705

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,f.par.pls  = 4)


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

`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):5117-5122. doi: 10.1073/pnas.1220873111. doi:10.1073/pnas.1220873111

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. doi:10.1093/bioinformatics/btw705

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)
```

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):5117-5122. doi: 10.1073/pnas.1220873111. doi:10.1073/pnas.1220873111

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. doi:10.1093/bioinformatics/btw705

See Also

swissroll,dinisurface,spirals

Examples

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

kabsch *Kabsch Algorithm*

Description

Aligns two sets of points via rotations and translations. Given two sets of points, with one specified as the reference set, the other set will be rotated so that the RMSD between the two is minimized. The format of the matrix is that there should be one row for each of n observations, and the number of columns, d, specifies the dimensionality of the points. The point sets must be of equal size and with the same ordering, i.e. point one of the second matrix is mapped to point one of the reference matrix, point two of the second matrix is mapped to point two of the reference matrix, and so on.

Usage

```
kabsch (pm, qm)
```

Arguments

pm	n x d matrix of points to align to to qm.
qm	n x d matrix of reference points.

Value

Matrix pm rotated and translated so that the ith point is aligned to the ith point of qm in the least-squares sense.

Author(s)

James Melville

Examples

```
data=iris[,-5]
pp1=pca(data)$x
pp2=pca(scale(data))$x
pp3=kabsch(pp1,pp2)
plot(pp1,pch=21,bg=rep(2:4,each=50))
points(pp3,pch=21,bg=rep(2:4,each=50),col=5)
```

KODAMA.matrix *Knowledge Discovery by Accuracy Maximization*

Description

Run KODAMA on a numeric data matrix and return the optimized label runs and nearest-neighbor structure used by KODAMA.visualization.

Usage

```
KODAMA.matrix(
  data,
  spatial = NULL,
  samples = NULL,
  M = 100,
  Tcycle = 20,
  ncomp = min(c(50, ncol(data))),
  W = NULL,
  metrics = "euclidean",
  constrain = NULL,
  fix = NULL,
  landmarks = 10000,
  splitting = ifelse(nrow(data) < 40000, 100, 300),
  spatial.resolution = 0.3,
  n.cores = 1,
  ancestry = FALSE,
  seed = 1234,
  ...
)
```

Arguments

data	Numeric matrix where rows are samples and columns are variables.
spatial	Optional numeric matrix of spatial coordinates with nrow(spatial) == nrow(data).
samples	Optional sample identifier vector used to separate multiple spatial samples on a shared coordinate axis.
M	Number of independent KODAMA optimization runs.
Tcycle	Number of optimization cycles for each run.
ncomp	Number of PLS components.
W	Optional starting labels for semi-supervised initialization.
metrics	Distance metric passed to Rnanoflann::nn.
constrain	Optional grouping constraint vector; entries with the same value are forced to share labels within each run.
fix	Optional logical vector indicating which entries in W are fixed during optimization.

landmarks	Number of landmark clusters used in each run.
splitting	Number of clusters used for initialization when W is NULL.
spatial.resolution	Fraction of landmarks used to define spatial constraint clusters.
n.cores	Number of worker processes. On Unix-like systems forked workers are used; on Windows PSOCK workers are used.
ancestry	Logical; if TRUE, ancestry-aware spatial processing is used.
seed	Random seed.
...	Ignored legacy arguments. Passing FUN is deprecated and has no effect.

Details

The function runs M independent KODAMA optimizations and builds a KODAMA-weighted nearest-neighbor structure. Progress bars are shown for both the optimization stage and dissimilarity update stage.

The PLS backend is selected automatically inside corecpp before each cross-validation step from the current number of classes: "plssvd" (fast mode) when ncomp is smaller than the number of classes, otherwise "simpls".

When n.cores > 1, Unix-like systems use fork-based parallelism, which typically reduces memory duplication through copy-on-write when worker code treats data as read-only. On Windows, socket workers are used and the input matrix is copied to workers by design.

Value

A list with:

acc	Numeric vector of length M with final run accuracies.
v	Numeric matrix (M x Tcycle) of accuracy trajectories.
res	Numeric matrix (M x nrow(data)) with optimized labels from each run.
knn_Rnanoflann	List containing indices, distances, and neighbors.
data	Input data matrix.
res_constrain	Numeric matrix (M x nrow(data)) with effective constraints used in each run.
n.cores	Number of cores used by KODAMA.matrix. This value is reused by KODAMA.visualization when visualization config sets define.n.cores = FALSE.

Author(s)

Stefano Cacciatore and Leonardo Tenori

See Also

KODAMA.visualization

Examples

```
data(iris)
data_mat <- iris[, -5]
kk <- KODAMA.matrix(data_mat, ncomp = 2, M = 10, n.cores = 1)
embedding <- KODAMA.visualization(kk, "t-SNE")
plot(embedding, col = as.numeric(iris[, 5]), 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("UMAP", "t-SNE"),
                     config=NULL)
```

Arguments

kk	output of KODAMA.matrix function, including n.cores.
method	method to be considered for transforming the dissimilarity matrix into a low-dimensional space. Choices are "t-SNE" and "UMAP".
config	object of class umap.config or tsne.config. If config$define.n.cores = FALSE (default), the number of threads is taken from kk$n.cores. If TRUE, the thread settings defined in config are used.

Value

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

Author(s)

Stefano Cacciatore and Leonardo Tenori

References

Abdel-Shafy EA, Kassim M, Vignol A, *et al.*
KODAMA enables self-guided weakly supervised learning in spatial transcriptomics.
bioRxiv 2025. doi: 10.1101/2025.05.28.656544. doi:10.1101/2025.05.28.656544

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-5122. doi: 10.1073/pnas.1220873111. doi:10.1073/pnas.1220873111

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. doi:10.1093/bioinformatics/btw705

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,ncomp=2)
cc=KODAMA.visualization(kk,"t-SNE")
plot(cc,col=as.numeric(labels),cex=2)
```

`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):5117-5122. doi: 10.1073/pnas.1220873111. doi:10.1073/pnas.1220873111

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. doi:10.1093/bioinformatics/btw705

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[,1:50]
plot(cc,pch=21,bg=class)

kk=KODAMA.matrix(cc,ncomp=2)

custom.settings=config.tsne.default
custom.settings$perplexity = 10
cc=KODAMA.visualization(kk,"t-SNE",config=custom.settings)

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

Abdel-Shafy EA, Kassim M, Vignol A, *et al.*
KODAMA enables self-guided weakly supervised learning in spatial transcriptomics.
bioRxiv 2025. doi: 10.1101/2025.05.28.656544. doi:10.1101/2025.05.28.656544

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.

Proc Natl Acad Sci U S A 2014;111(14):5117-5122. doi: 10.1073/pnas.1220873111. doi:10.1073/pnas.1220873111

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. doi:10.1093/bioinformatics/btw705

See Also

KODAMA.matrix,KODAMA.visualization

Examples

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

`MDS.defaults` *Default configuration for RMDS*

Description

A list with parameters customizing an MDS embedding.

Usage

```
MDS.defaults
```

Format

An object of class `MDS.defaults` of length 1.

Details

dims: integer, Output dimensionality

Examples

```
# display all default settings
MDS.defaults

# create a new settings object with perplexity set to 100
custom.settings = MDS.defaults
custom.settings$dims = 3
custom.settings
```

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. doi:10.1073/pnas.0705685105

Cacciatore S, Luchinat C, Tenori L
Knowledge discovery by accuracy maximization.
Proc Natl Acad Sci U S A 2014;111(14):5117-5122. doi: 10.1073/pnas.1220873111. doi:10.1073/pnas.1220873111

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. doi:10.1093/bioinformatics/btw705

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$donor))

u_pca=pca(u)$x[,1:5]

kk=KODAMA.matrix(u_pca,ncomp=2)
cc=KODAMA.visualization(kk,"t-SNE")
plot(cc,pch=21,bg=rainbow(22)[class])
```

`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):5117-5122. doi: 10.1073/pnas.1220873111. doi:10.1073/pnas.1220873111

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. doi:10.1093/bioinformatics/btw705

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 *Truncated Principal Components Analysis*

Description

Performs PCA using KODAMA's internal vendored IRLBA backend (with a small-matrix `svd()` fallback) and returns a `prcomp`-compatible object.

Usage

```
pca(x, nv = min(50L, ncol(x)), ...)
```

Arguments

`x`	A numeric matrix of data.
`nv`	Number of principal components to compute.
`...`	Currently unused, kept for backward compatibility.

Value

The function returns a list with class `prcomp` containing:

`sdev`	standard deviations of the retained principal components.
`rotation`	matrix of variable loadings (columns are retained components).
`x`	scores matrix equivalent to `u %*% diag(d)` from truncated SVD.
`center`, `scale`	set to `FALSE`; centering/scaling are expected upstream when needed.
`txt`	percentage-of-variance labels for each retained component.

Author(s)

Stefano Cacciatore

References

Baglama J, Reichel L.
Augmented implicitly restarted Lanczos bidiagonalization methods.
SIAM Journal on Scientific Computing 2005;27(1):19-42.

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, nv = 5)
plot(cc$x, pch = 21, bg = class)
```

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):5117-5122. doi: 10.1073/pnas.1220873111. doi:10.1073/pnas.1220873111

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. doi:10.1093/bioinformatics/btw705

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):5117-5122. doi: 10.1073/pnas.1220873111. doi:10.1073/pnas.1220873111

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. doi:10.1093/bioinformatics/btw705

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):5117-5122. doi: 10.1073/pnas.1220873111. doi:10.1073/pnas.1220873111

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. doi:10.1093/bioinformatics/btw705

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, 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. doi:10.1093/bioinformatics/btw705

Examples

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

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):5117-5122. doi: 10.1073/pnas.1220873111. doi:10.1073/pnas.1220873111

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. doi:10.1093/bioinformatics/btw705

Examples

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

data(USA)

pp=pca(USA$data)$x[,1:50]

kk=KODAMA.matrix(pp,ncomp=2)
custom.settings=config.tsne.default
custom.settings$perplexity = 10
cc=KODAMA.visualization(kk,"t-SNE",config=custom.settings)
oldpar <- par(cex=0.5,mar=c(15,6,2,2));
plot(USA$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

∗ **datasets**
  config.tsne.default, 2
  config.umap.default, 3
  lymphoma, 15
  MDS.defaults, 17
  MetRef, 18
  USA, 26
∗ **dataset**
  dinisurface, 7
  helicoid, 9
  spirals, 23
  swissroll, 24
∗ **normalization**
  normalization, 19
∗ **pca**
  kabsch, 10
  pca, 21
∗ **scaling**
  scaling, 22
∗ **transformation**
  transformy, 25

config.tsne.default, 2
config.umap.default, 3
core_cpp, 5

dinisurface, 7, *9*, *24, 25*

floyd, 8

helicoid, *7*, 9, *24, 25*

kabsch, 10
KODAMA.matrix, *3, 4*, *6*, 11, *13*, *17*
KODAMA.visualization, *3, 4*, *6*, *11, 12*, 13, *14*, *17*

lymphoma, 15

mcplot, 16
MDS.defaults, 17
MetRef, 18

normalization, 19, *23*

pca, 21

scaling, *20*, 22
spirals, *7*, *9*, 23, *25*
swissroll, *7*, *9*, 24, *24*

transformy, 25

USA, 26