Package ‘opticskxi’

March 9, 2025

Title OPTICS K-Xi Density-Based Clustering

Version 1.2.1

Description Density-based clustering methods are well adapted to the clustering of high-dimensional data and enable the discovery of core groups of various shapes despite large amounts of noise. This package provides a novel density-based cluster extraction method, OPTICS k-Xi, and a framework to compare k-Xi models using distance-based metrics to investigate datasets with unknown number of clusters. The vignette first introduces density-based algorithms with simulated datasets, then presents and evaluates the k-Xi cluster extraction method. Finally, the models comparison framework is described and experimented on 2 genetic datasets to identify groups and their discriminating features. The k-Xi algorithm is a novel OPTICS cluster extraction method that specifies directly the number of clusters and does not require fine-tuning of the steepness parameter as the OPTICS Xi method. Combined with a framework that compares models with varying parameters, the OPTICS k-Xi method can identify groups in noisy datasets with unknown number of clusters. Results on summarized genetic data of 1,200 patients are in Charlon T. (2019) <doi:10.13097/archive-ouverte/unige:161795>. A short video tutorial can be found at <https://www.youtube.com/watch?v=P2XAjqI5Lc4/>.

Imports ggplot2, magrittr, Matrix, rlang

Depends R (>= 3.5.0)

Suggests amap, dbscan, cowplot, fastICA, fpc, ggrepel, grid, grDevices, gtable, knitr, parallel, plyr, reshape2, testthat

VignetteBuilder knitr

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

URL https://gitlab.com/thomaschln/opticskxi

BugReports https://gitlab.com/thomaschln/opticskxi/-/issues

NeedsCompilation no

Author Thomas Charlon [aut, cre] (<https://orcid.org/0000-0001-7497-0470>)

Maintainer Thomas Charlon <charlon@protonmail.com>

Repository CRAN

Date/Publication 2025-03-09 22:50:02 UTC

Contents

`contingency_table` *Contingency table*

Description

Include NAs and add totals to table.

Usage

```
contingency_table(...)
```

Arguments

`...`	Passed to table

Value

Table object

cosine_simi *Cosine similarity between vectors and/or matrices.*

Description

Inputs will be L2 normalized, then matrix multiplied (y is transposed). If second input is missing, first input will be recycled, which enables to efficiently compute cosine similarities between the rows of a rectangular matrix.

Usage

```
cosine_simi(x, y)
```

Arguments

x	Numeric vector or matrix
y	Numeric vector or matrix. If missing, copied from parameter x.

Value

Symmetric numeric similarity matrix

crohn *Crohn's disease data*

Description

The data set consist of 103 common (>5% minor allele frequency) SNPs genotyped in 129 trios from an European-derived population. These SNPs are in a 500-kb region on human chromosome 5q31 implicated as containing a genetic risk factor for Crohn disease.

Imported from the gap R package.

An example use of the data is with the following paper, Kelly M. Burkett, Celia M. T. Greenwood, BradMcNeney, Jinko Graham. Gene genealogies for genetic association mapping, with application to Crohn's disease. Fron Genet 2013, 4(260) doi: 10.3389/fgene.2013.00260

Usage

```
data(crohn)
```

Format

A data frame containing 387 rows and 212 columns

Source

MJ Daly, JD Rioux, SF Schaffner, TJ Hudson, ES Lander (2001) High-resolution haplotype structure in the human genome Nature Genetics 29:229-232

dist_matrix *dist_matrix*

Description

Dispatch of amap::Dist, cosine_dist, and norm_inprod methods.

Usage

```
dist_matrix(data, method = "euclidean", n_cores = 1)
```

Arguments

data	Rectangular numeric matrix [Observations, Features]
method	Methods accepted by amap::Dist or cosine and norm_inprod
n_cores	Number of cores

Value

Distance symmetric matrix

ensemble_metrics *Compute ensemble metrics*

Description

Use models' rankings over several metrics to select best model. Several approaches can be taken to sum the models' rankings, and instead of summing the ranks of all models over all metrics, we prefer to rank only the top models for each metrics, and set 0 to all other. This behavior is controlled by the n_top parameter. In a second step, we sum the ranks and return only the top models, and this is controlled by the n_models parameter. The output is a list of the rankings matrix, for quality control purposes, and the selected models' parameters data frame, which is used by the ensemble_models function.

Usage

```
ensemble_metrics(
  n_top = 0,
  df_params,
  metrics = NULL,
  metrics_exclude = NULL,
  n_models = 10
)
```

Arguments

`n_top`	Threshold of number of models to rank
`df_params`	Output of opticskxi_pipeline
`metrics`	Names of metrics to use. Any of those computed by opticskxi_pipeline, e.g. 'sindex', 'ch', 'dunn', 'dunn2', 'widestgap', 'entropy' etc. NULL for all (8).
`metrics_exclude`	Names of metrics to exclude. Typically used with metrics = NULL. E.g. 'entropy'.
`n_models`	Number of best models to return

Value

List of metrics' rankings matrix and best models' parameters data frame.

`ensemble_metrics_bootstrap` *Select models based on ensemble metrics*

Description

Typically we will call ensemble_metrics with varying numbers of ranks to consider and this function will sum up the ranks from those calls.

Usage

```
ensemble_metrics_bootstrap(l_ensemble_metrics, n_models = 4)
```

Arguments

`l_ensemble_metrics`	Output of function ensemble_metrics
`n_models`	Number of best models to return

Value

List of parameters of best models

ensemble_models *Select best models based on ensemble metrics*

Description

Call ensemble_metrics with varying numbers of rank thresholds to consider and sum up the ranks from those calls.

Usage

```
ensemble_models(
  df_kxi,
  n_models = 4,
  metrics = NULL,
  metrics_exclude = NULL,
  model_subsample = c(0.1, 0.2, 0.5),
  n_models_subsample = 10
)
```

Arguments

`df_kxi` Output of opticskxi_pipeline function. Dataframe with models' parameters and OPTICS k-Xi results

`n_models` Number of best models to return

`metrics` Names of metrics to use. Any of those computed by opticskxi_pipeline, e.g. 'sindex', 'ch', 'dunn', 'dunn2', 'widestgap', 'entropy' etc. NULL for all (8).

`metrics_exclude` Names of metrics to exclude. Typically used with metrics = NULL. E.g. 'entropy'.

`model_subsample` Ratios of best models to consider.

`n_models_subsample` Number of best models when subsampling.

Value

Input object df_kxi subsetted to best models according to ensemble metrics.

Examples

```
data('m_psych_embeds')
m_psych_embeds = m_psych_embeds[1:200, 1:20]

df_params = expand.grid(n_xi = 4:5, pts = c(5, 10), dist = 'cosine',
                        dim_red = 'ICA', n_dimred_comp = 5)

df_kxi = opticskxi_pipeline(m_psych_embeds, df_params,
                                metrics_dist = 'cosine',
                                n_min_clusters = 2, n_cores = 1,
                                metrics = c('avg.silwidth', 'dunn'))

df_kxi = ensemble_models(df_kxi, n_models = 4,
                         model_subsample = c(0.4, 0.6),
                         n_models_subsample = 4)
```

fortify_dimred *Fortify a dimension reduction object*

Description

Fortify a dimension reduction object

Usage

```
fortify_dimred(
  m_dimred,
  m_vars = NULL,
  v_variance = NULL,
  sup_vars = NULL,
  var_digits = 1
)
```

Arguments

m_dimred	Projection matrix
m_vars	Rotation matrix (optional)
v_variance	Explained variance (optional)
sup_vars	Optional supplementary variables
var_digits	Explained variance percent digits

Value

Data frame

See Also

fortify_pca, fortify_ica

Examples

```
pca <- prcomp(iris[-5])
df_pca <- fortify_dimred(pca$x)
```

fortify_ica *Get and fortify ICA*

Description

Get and fortify ICA

Usage

```
fortify_ica(m_data, ..., sup_vars = NULL)
```

Arguments

m_data	Input matrix
...	Passed to fastICA::fastICA
sup_vars	Optional supplementary variables

Value

Fortified dimension reduction

See Also

fortify_dimred, fortify_pca

Examples

```
df_ica <- fortify_ica(iris[-5], n.comp = 2)
```

fortify_pca *Get and fortify PCA*

Description

Get and fortify PCA

Usage

```
fortify_pca(m_data, ..., sup_vars = NULL)
```

Arguments

m_data	Input matrix
...	Passed to stats::prcomp
sup_vars	Optional supplementary variables

Value

Fortified dimension reduction

See Also

fortify_dimred, fortify_ica

Examples

```
df_pca <- fortify_pca(iris[-5])
df_pca <- fortify_pca(iris[-5], sup_vars = iris[5])
```

---

get_best_kxi *Get best k-Xi model*

Description

Select k-Xi clustering model based on a metric and a rank

Usage

```
get_best_kxi(df_kxi, metric = "avg.silwidth", rank = 1)
```

Arguments

`df_kxi`	Data frame returned by opticsxi_pipeline
`metric`	Metric to choose best model
`rank`	Rank(s) of model to choose, ordered by decreasing metric

Value

df_kxi row with specified metric and rank, simplified to a list if only one rank selected

See Also

opticskxi_pipeline

ggpairs — *Plot multiple axes of a data frame or a fortified dimension reduction.*

Description

Plot multiple axes of a data frame or a fortified dimension reduction.

Usage

```
ggpairs(
  df_data,
  group = NULL,
  axes = 1:2,
  variables = FALSE,
  n_vars = 0,
  ellipses = FALSE,
  ...,
  title = NULL,
  colors = if (!is.null(group)) nice_palette(df_data[[group]])
)
```

Arguments

`df_data`	Data frame
`group`	Column name of the grouping of observations
`axes`	Axes to plot. If more than 2, plots all pair combinations
`variables`	Logical, plot variable contributions of the dimension reduction to the selected axes, only for 2 axes
`n_vars`	Maximum number of variable contributions to plot. By default 0, for all variables.
`ellipses`	Logical, plot ellipses of groups
`...`	Passed to ggplot2 stat_ellipse if ellipses are requested
`title`	String to add as title, default NULL
`colors`	Vector of colors for each group

Value

ggmatrix

See Also

fortify_pca, fortify_ica

Examples

```
df_pca <- fortify_pca(iris[-5])
ggpairs(df_pca)
df_pca <- fortify_pca(iris[-5], sup_vars = iris[5])
ggpairs(df_pca, group = 'Species', ellipses = TRUE, variables = TRUE)
```

`ggplot_kxi_metrics` *Ggplot OPTICS k-Xi metrics*

Description

Plot metrics of a kxi_pipeline output

Usage

```
ggplot_kxi_metrics(df_kxi, metric = c("avg.silwidth", "bw.ratio"), n = 8)
```

Arguments

`df_kxi`	Data frame returned by opticskxi_pipeline
`metric`	Vector of metrics to display from the df_kxi object
`n`	Number of best models for the first metric to display

Value

ggplot

See Also

opticskxi_pipeline

`ggplot_optics` *Ggplot optics*

Description

Plot OPTICS reachability plot.

Usage

```
ggplot_optics(
  optics_obj,
  groups = NULL,
  colors = if (!is.null(groups)) nice_palette(groups),
  segment_size = 300/nrow(df_optics)
)
```

Arguments

`optics_obj`	dbscan::optics object
`groups`	Optional vector defining groups of OPTICS observations
`colors`	If groups specified, vector of colors for each group
`segment_size`	Size for geom_segment

Value

ggplot

See Also

opticskxi

Examples

```
data('multishapes')
optics_obj <- dbscan::optics(multishapes[1:2])
ggplot_optics(optics_obj)
ggplot_optics(optics_obj,
  groups = opticskxi(optics_obj, n_xi = 5, pts = 30))
```

`gtable_kxi_profiles` *Gtable OPTICS k-Xi distance profiles*

Description

Plot OPTICS distance profiles of k-Xi clustering models

Usage

```
gtable_kxi_profiles(df_kxi, metric = "avg.silwidth", rank = 1:4, ...)
```

Arguments

`df_kxi`	Data frame returned by opticskxi_pipeline
`metric`	Metric to choose best clustering model
`rank`	Ranks of models to plot, ordered by decreasing model metric
`...`	Passed to ggplot_kxi_profile

See Also

opticskxi_pipeline

`hla` *The HLA data*

Description

This data set contains HLA markers DRB, DQA, DQB and phenotypes of 271 Schizophrenia patients (y=1) and controls (y=0). Genotypes for 3 HLA loci have prefixes name (e.g., "DQB") and a suffix for each of two alleles (".a1" and ".a2").

Imported from the gap package.

Usage

```
data(hla)
```

Format

A data frame containing 271 rows and 8 columns

Source

Dr Padraig Wright of Pfizer

`multishapes` *A dataset containing clusters of multiple shapes*

Description

Data containing clusters of any shapes. Useful for comparing density-based clustering (DBSCAN) and standard partitioning methods such as k-means clustering. Imported from the factoextra package.

Usage

```
data("multishapes")
```

Format

A data frame with 1100 observations on the following 3 variables.

`x` a numeric vector containing the x coordinates of observations

`y` a numeric vector containing the y coordinates of observations

`shape` a numeric vector corresponding to the cluster number of each observations.

Details

The dataset contains 5 clusters and some outliers/noises.

Examples

```
data('multishapes')
plot(multishapes[, 1], multishapes[, 2],
    col = multishapes[, 3], pch = 19, cex = 0.8)
```

m_psych_embeds *A dataset containing the embeddings matrix of psychological related words*

Description

Data containing Glove embeddings of psychological related words, useful for demonstrating the use of ensemble metrics.

Usage

```
data("m_psych_embeds")
```

Format

A matrix with 831 words in rows and 100 embedding dimensions in columns.

Details

The dataset contains groups of related words among other irrelevant words.

nice_palette *Nice palette*

Description

Color palette

Usage

```
nice_palette(groups, rainbow = FALSE)
```

Arguments

`groups`	Vector, each unique value will get a color
`rainbow`	If TRUE, rainbow-like colors, else differentiate successive values

Value

Vector of colors

normalize *Matrix normalization*

Description

normalize matrix rows using given norm. Copied from text2vec

Usage

```
normalize(m, norm = c("l1", "l2", "none"))
```

Arguments

`m`	`matrix` (sparse or dense).
`norm`	`character` the method used to normalize term vectors

Value

normalized matrix

norm_inprod *norm_inprod*

Description

Normalized inner product with transposed input matrix

Usage

```
norm_inprod(m)
```

Arguments

`m`	Numeric matrix

Value

Numeric matrix

`opticskxi` *OPTICS k-Xi clustering algorithm*

Description

For each largest distance differences on the OPTICS profile, consecutive observations left and right on the OPTICS profile (i.e. lower and higher OPTICS id) will be assigned to 2 different clusters if their distance is below the distance of the edge point. If above, observations are NA. The pts parameter defines a minimum number of observations to form a valley (i.e. cluster). If the number of observations in one valley is smaller than pts, observations are set to NA.

Usage

```
opticskxi(
  optics_obj,
  n_xi,
  pts = optics_obj$minPts,
  max_loop = 50,
  verbose = FALSE
)
```

Arguments

optics_obj	Data frame returned by optics
n_xi	Number of clusters to define
pts	Minimum number of points per clusters
max_loop	Maximum iterations to find n_xi clusters
verbose	Print the ids of the largest difference considered and cluster information if they define one

Value

Vector of clusters

See Also

opticskxi_pipeline, ggplot_optics

Examples

```
data('multishapes')
optics_shapes <- dbscan::optics(multishapes[1:2])
kxi_shapes <- opticskxi(optics_shapes, n_xi = 5, pts = 30)
ggplot_optics(optics_shapes, groups = kxi_shapes)
ggpairs(cbind(multishapes[1:2], kXi = kxi_shapes), group = 'kXi')
```

opticskxi_pipeline *OPTICS k-Xi models comparison pipeline*

Description

Computes OPTICS k-Xi models based on a parameter grid, binds results in a data frame, and computes distance based metrics for each model.

Usage

```
opticskxi_pipeline(
  m_data,
  df_params = expand.grid(n_xi = 1:10, pts = c(20, 30, 40), dist = c("euclidean",
   "abscorrelation"), dim_red = c("identity", "PCA", "ICA"), n_dimred_comp = c(5, 10,
    20)),
  metrics_dist = c("euclidean", "cosine"),
  max_size_ratio = 1,
  n_min_clusters = 0,
  n_cores = 1,
  ...
)
```

Arguments

`m_data`	Data matrix
`df_params`	Parameter grid for the OPTICS k-Xi function call and optional dimension reduction. Required columns: n_xi, pts, dist. Optonal columns: dim_red, n_dim_red.
`metrics_dist`	Distance used for metrics, either euclidean or cosine.
`max_size_ratio`	Maximum size ratio of clusters. E.g. for 0.8, if a cluster is larger than 80% of points it will be removed.
`n_min_clusters`	Minimum number of clusters. Ignored if 0.
`n_cores`	Number of cores
`...`	Passed to get_kxi_metrics

Value

Input parameter data frame with with results binded in columns optics, clusters and metrics.

See Also

get_best_kxi, ggplot_kxi_metrics, gtable_kxi_profiles

Examples

```
data('hla')
m_hla <- hla[-c(1:2)] %>% scale

df_params_hla <- expand.grid(n_xi = 3:5, pts = c(20, 30),
  dist = c('manhattan', 'euclidean'))

df_kxi_hla <- opticskxi_pipeline(m_hla, df_params_hla)

ggplot_kxi_metrics(df_kxi_hla, n = 8)
gtable_kxi_profiles(df_kxi_hla) %>% plot

best_kxi_hla <- get_best_kxi(df_kxi_hla, rank = 2)
clusters_hla <- best_kxi_hla$clusters

fortify_pca(m_hla, sup_vars = data.frame(Clusters = clusters_hla)) %>%
  ggpairs('Clusters', ellipses = TRUE, variables = TRUE)
```

`print_vignette_table` *Print vignette table*

Description

Print knitr::kable latex table with legend at bottom.

Usage

```
print_vignette_table(table_obj, label)
```

Arguments

`table_obj`	Table object
`label`	Latex label

Value

None, side-effect prints a Latex table

`residuals_table` *Residuals table*

Description

Bind contingency table and Pearson Chi-squared residuals.

Usage

```
residuals_table(...)
```

Arguments

`...`	Passed to contingency_table and chisq.test

Value

Matrix

`stddev_mean` *stddev_mean*

Description

Get mean of standard deviations of matrix columns

Usage

```
stddev_mean(m)
```

Arguments

`m`	Numeric matrix

Value

Mean of standard deviations of matrix columns

%<>% *Assignment pipe*

Description

Pipe an object forward into a function or call expression and update the ‘lhs‘ object with the resulting value. Magrittr imported function, see details and examples in the magrittr package.

Arguments

`lhs`	An object which serves both as the initial value and as target.
`rhs`	a function call using the magrittr semantics.

Value

None, used to update the value of lhs.

%$% *Exposition pipe*

Description

Expose the names in ‘lhs‘ to the ‘rhs‘ expression. Magrittr imported function, see details and examples in the magrittr package.

Arguments

`lhs`	A list, environment, or a data.frame.
`rhs`	An expression where the names in lhs is available.

Value

Result of rhs applied to one or several names of lhs.

%>% *Pipe*

Description

Pipe an object forward into a function or call expression. Magrittr imported function, see details and examples in the magrittr package.

Arguments

`lhs`	A value or the magrittr placeholder.
`rhs`	A function call using the magrittr semantics.

Value

Result of rhs applied to lhs, see details in magrittr package.

Index