

# Package ‘opticskxi’

October 14, 2022

**Title** OPTICS K-Xi Density-Based Clustering

**Version** 0.1

**Author** Thomas Charlon <charlon@protonmail.com>

**Maintainer** Thomas Charlon <charlon@protonmail.com>

**Description** 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.

**Imports** ggplot2, magrittr

**Depends** R (>= 2.15)

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

**VignetteBuilder** knitr

**License** GPL-3 | file LICENSE

**Encoding** UTF-8

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-07-18 22:30:02 UTC

## R topics documented:

contingency_table . . . . .	2
crohn . . . . .	2
fortify_dimred . . . . .	3
fortify_ica . . . . .	4
fortify_pca . . . . .	4
get_best_kxi . . . . .	5
ggpairs . . . . .	6
ggplot_kxi_metrics . . . . .	7
ggplot_optics . . . . .	7

gtable_kxi_profiles . . . . .	8
hla . . . . .	9
multishapes . . . . .	9
nice_palette . . . . .	10
opticskxi . . . . .	10
opticskxi_pipeline . . . . .	11
print_table . . . . .	12
residuals_table . . . . .	13
%<% . . . . .	13
%%\$% . . . . .	13
%>% . . . . .	13

**Index** **14**

---

contingency_table	<i>Contingency table</i>
-------------------	--------------------------

---

**Description**

Include NAs and add totals to table.

**Usage**

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

**Arguments**

... Passed to table

**Value**

Table object

---

crohn	<i>Crohn's disease data</i>
-------	-----------------------------

---

**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

---

fortify_dimred	<i>Fortify a dimension reduction object</i>
----------------	---

---

**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	<i>Get and fortify ICA</i>
-------------	----------------------------

---

**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	<i>Get and fortify PCA</i>
-------------	----------------------------

---

**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	<i>Get best k-Xi model</i>
--------------	----------------------------

---

**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**

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

[opticksxi\\_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**

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

**See Also**

[opticskxi\\_pipeline](#)

---

hla	<i>The HLA data</i>
-----	---------------------

---

**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	<i>A dataset containing clusters of multiple shapes</i>
-------------	---

---

**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)
```

---

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

---

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)),
  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. Optional columns: dim_red, n_dim_red.
n_cores	Number of cores

**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\_table

*Print table*

---

**Description**

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

**Usage**

```
print_table(table_obj, label)
```

**Arguments**

table_obj	Table object
label	Latex label

**Value**

None

---

residuals_table	<i>Residuals table</i>
-----------------	------------------------

---

**Description**

Bind contingency table and Pearson Chi-squared residuals.

**Usage**

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

**Arguments**

... Passed to contingency\_table and chisq.test

**Value**

Matrix

---

<code>%&lt;&gt;%</code>	<i>Magrittr pipe-assign operator</i>
-------------------------	--------------------------------------

---

**Description**

Magrittr pipe-assign operator

---

<code>%%\$%</code>	<i>Magrittr pipe-with operator</i>
--------------------	------------------------------------

---

**Description**

Magrittr pipe-with operator

---

<code>%&gt;%</code>	<i>Magrittr pipe operator</i>
---------------------	-------------------------------

---

**Description**

Magrittr pipe operator

# Index

## \* datasets

crohn, 2  
hla, 9

%<>%, 13

%>%, 13

%\$%, 13

contingency\_table, 2

crohn, 2

fortify\_dimred, 3, 4, 5

fortify\_ica, 3, 4, 5, 6

fortify\_pca, 3, 4, 4, 6

get\_best\_kxi, 5, 12

ggpairs, 6

ggplot\_kxi\_metrics, 7, 12

ggplot\_optics, 7, 11

gtable\_kxi\_profiles, 8, 12

hla, 9

multishapes, 9

nice\_palette, 10

opticskxi, 8, 10

opticskxi\_pipeline, 5, 7, 8, 11, 11

print\_table, 12

residuals\_table, 13