Package 'tidydr'

October 9, 2026

Title Unify Dimensionality Reduction Results

Version 0.0.7

Description Dimensionality reduction is widely used in many domains for analyzing and visualizing high-dimensional data. 'tidydr' provides uniform output and is compatible with multiple methods, including 'prcomp', 'cmdscale', 'Rtsne', 'umap' and 'metaMDS'. Any function returning a numeric matrix can also be used. The unified result can be visualized directly with 'ggplot2', and several methods can be run and compared in a single call.

Imports cluster, ggfun, ggplot2, grid, rlang, stats, utils

Suggests knitr, rmarkdown, prettydoc, SingleCellExperiment, SummarizedExperiment, ape, MASS, Rtsne, uwot, vegan, smacof, ecodist, ade4, labdsv, testthat (>= 3.0.0)

VignetteBuilder knitr

ByteCompile true

License Artistic-2.0

URL https://github.com/YuLab-SMU/tidydr/

BugReports https://github.com/YuLab-SMU/tidydr/issues

Encoding UTF-8

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Guangchuang Yu [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-6485-8781>), Shuangbin Xu [aut] (ORCID: <https://orcid.org/0000-0003-3513-5362>)

Maintainer Guangchuang Yu <guangchuangyu@gmail.com>

Repository CRAN

Date/Publication 2026-10-09 12:30:02 UTC

Contents

available_methods 2
dr 3
dr_compare 4
dr_extract 5
element_line2 6
nk 7
silinfo_widths 9
theme_dr 10
theme_noaxis 10

Index **12**

`available_methods` *List dimensionality reduction methods currently available*

Description

This function shows available methods that worked for `dr()` function.

Usage

```
available_methods(method = "all")
```

Arguments

`method`	one of 'data', 'distance' or 'all' (default)

Value

A character vector of available DR methods

Author(s)

Lang Zhou and Guangchuang Yu

Examples

```
available_methods()
```

dr *dr*

Description

dimensional reduction

Usage

```
dr(data, fun, ...)
```

Arguments

data	input data
fun	function to perform dimensional reduction
...	additional parameters passed to 'fun'

Details

This function call the user-provided function ('fun') to perform dimensional reduction on the input data ('data')

`sample_info` is an optional list that a `dr_extract()` method may return alongside the coordinates. It is meant to carry method-specific, per-sample information (e.g. local density, k-nearest-neighbour indices or cluster labels). Every element of `sample_info` that holds one value per sample – a vector or factor of length `nrow(data)` – is merged into the data.frame returned by `fortify()`, using the element name as the column name, so that it can be mapped in `autoplot()`. Elements that do not hold one value per sample, and elements whose name is already taken by a coordinate or a metadata column, are not merged and a warning is issued. None of the `dr_extract()` methods shipped with the package returns a `sample_info` element, so `sample_info` is `NULL` unless a custom `dr_extract()` method supplies one.

Value

a DrResult object, which contains 'data' (original data), 'drdata' (coordination after dimensionality reduction), eigenvalue (standard deviation explained by each dimension), stress (evaluate the effect of dimensionality reduction) and 'sample_info' (method-specific additional information, see Details)

Author(s)

Guangchuang Yu

Examples

```
x = dr(iris[,1:4], prcomp)
autoplot(x, aes(color=.group), metadata=iris$Species)
```

dr_compare *dr_compare*

Description

Compare several dimensionality reduction methods

Usage

```
dr_compare(
  data,
  funs = list(prcomp = stats::prcomp),
  dim = 1:2,
  metadata = NULL,
  ...
)
```

Arguments

data	input data. It follows the same contract as dr(): a numeric matrix, a numeric data.frame, or a 'dist' object.
funs	a named list of methods to compare. Each element is either a function, or a list with components fun (the function) and args (an optional list of method-specific arguments). The names are used as the method labels in summary and in the plot, so they must be unique.
dim	the two dimensions to extract, as a numeric vector of length 2. Defaults to 1:2, i.e. the first two dimensions.
metadata	optional sample-level metadata. It is forwarded to fortify() when the long table used by autoplot() is built, so that it becomes available as a column of the plot.
...	additional arguments passed to every method in funs. Arguments supplied through the args element of a method take precedence over ... for that method.

Details

dr_compare() applies several dimensionality reduction (DR) methods to the same input data and collects their results, so that the methods can be compared side by side. Every method is evaluated independently: a method that fails, or that returns fewer dimensions than requested, does not interrupt the others. The outcome of every method is reported in the summary component, which is the main purpose of this function.

A method that fails is never given placeholder coordinates. The failure is recorded in summary$error, and the corresponding element of results is the condition (error) object, so it can be inspected further.

Value

A DrCompare object, a list with the following components:

results a named list with one element per method. For a method that ran successfully it is a DrResult (see dr()); for a method that failed it is the condition (error) object.

summary a data.frame with one row per method and the columns method, status, n, k, has_eigenvalue, has_stress and error. status is "ok" if the method ran and returned at least max(dim) dimensions, "insufficient_dims" if it ran but returned too few dimensions, and "failed" if it raised an error. n and k are the number of samples and the number of extracted dimensions, has_eigenvalue and has_stress indicate whether the method produced those quantities, and error holds the error message of a failed method (NA otherwise).

dim, metadata, funs the corresponding inputs, kept so that autoplot() can rebuild the plot.

Author(s)

Guangchuang Yu

See Also

dr(), available_methods()

Examples

```
x <- dr_compare(iris[, 1:4], list(pca = stats::prcomp))
x$summary
autoplot(x)
```

---

dr_extract *dr_extract*

Description

dr_extract generic

Usage

```
dr_extract(result)
```

Arguments

result DrResult object

Details

`dr_extract()` is an S3 generic. Methods are provided for the result classes of the supported dimensionality reduction functions. In addition, a fallback method is provided for plain numeric matrices: any function passed to `dr()` that returns a numeric matrix with at least two columns is accepted, and its columns are used as the reduced coordinates.

The list returned by a method must contain `drdata`, the reduced coordinates: either a data.frame, or a numeric matrix, with one row per sample and at least two columns (one per retained dimension). A matrix is converted to a data.frame, so that `fortify()` – a 'ggplot2' generic – always returns a data.frame. The columns are renamed `Dim1`, `Dim2`, ... by `dr()`, and a `drdata` that is neither a data.frame nor a numeric matrix is rejected.

`eigenvalue`, `stress` and `sample_info` are optional. `sample_info` is a list of method-specific, per-sample vectors (e.g. local density or cluster labels), which `fortify()` merges into its result as extra columns; see `dr()` for the details.

Value

a list that contains components to construct a 'DrResult' object.

Author(s)

Guangchuang Yu

`element_line2` *element_line2*

Description

element_line2 for drawing shorten axis lines

Usage

```
element_line2(
  colour = NULL,
  size = NULL,
  linetype = NULL,
  lineend = NULL,
  color = NULL,
  arrow = NULL,
  inherit.blank = FALSE,
  id,
  xlength = 0.3,
  ylength = 0.3,
  ...
)
```

Arguments

colour	line colour
size	line size in pts
linetype	line type
lineend	line end style (round, butt, square)
color	aliase to colour
arrow	arrow specification, as created by 'grid::arrow()'
inherit.blank	whether inherit 'element_blank'
id	1 or 2, 1 for axis.line.x.bottom and 2 for axis.line.y.left, only these two axes supported
xlength	length of x axis
ylength	length of y axis
...	additional parameters

Value

element_line2 object, which is a tailored element_line object

Author(s)

Guangchuang Yu

---

nk *nk*

Description

Choose best K (number of clusters)

Usage

```
nk(data, k, fun = pam, ...)
```

Arguments

data	input data (a matrix, data frame, or 'dist' object)
k	a vector of candidate number of clusters
fun	a function to perform the clustering. It must accept the number of clusters as its second argument, or through an argument named `k` or `centers` (e.g. `cluster::pam`, `stats::kmeans`). A method that takes no cluster count (e.g. `stats::hclust`) is called on the dissimilarity and cut with `stats::cutree()`. `fun` must return either a vector of cluster labels of length `nrow(data)`, or an object labels can be extracted from (`$clustering`, `$cluster` or `$labels`). Defaults to `cluster::pam`.

Whether fun is given a cluster count is decided by inspecting its arguments: a k or centers argument means yes, a first argument named d (the stats::hclust() convention) means no. A method that takes a dissimilarity as a first argument under some other name, such as cluster::agnes() or cluster::diana(), is therefore read as taking a cluster count and will fail; wrap it so that k is an argument of its own, e.g. nk(x, 2:4, fun = function(z, k) stats::cutree(cluster::agnes(stats::di
k)).

... additional parameters passed to fun

Note: by default this function calls cluster::pam() once for each candidate k, and pam() has O(n^2) time and memory complexity in the number of samples. For large datasets (e.g., thousands of cells/samples), this can be slow or memory-intensive. Consider subsampling the data or reducing the range of k for exploratory use.

Details

This function calculate the silhouette scores of each K (number of clusters). The output object can be used to choose the best K (via summary() or autoplot() methods)

The silhouette width is computed with cluster::silhouette() from the cluster labels and the dissimilarity of data (stats::dist(data), unless data is already a 'dist' object, in which case it is used as is). With the default fun = cluster::pam the silhouette information that pam() computes itself is reused verbatim, so the default result is unchanged.

Each element of $silinfo follows the shape produced by cluster::pam(): widths (a matrix with columns cluster, neighbor and sil_width), clus.avg.widths (per-cluster averages) and avg.width (the overall average). The row order of widths is the one produced by the clusterer, so it should be aligned by sample name rather than by position.

Value

a silinfo object, which contains 'data' (original data), 'silinfo' (silhouette scores), and k (the input k vector)

Author(s)

Guangchuang Yu

Examples

```
x <- nk(iris[,-5], 2:8)
summary(x)
# to visualize the average silhouete score (y axis) with k (x axis)
autoplot(x)
# to visualize a PCA plot color by the choosing k
autoplot(x, k=3)

# another clustering method
x2 <- nk(iris[,-5], 2:4, fun = stats::kmeans)
summary(x2)
# a hierarchical method is cut automatically
```

```
x3 <- nk(iris[,-5], 2:4, fun = stats::hclust)
summary(x3)
```

`silinfo_widths` *silinfo_widths*

Description

Per-sample silhouette widths

Usage

```
silinfo_widths(x, k)
```

Arguments

x	a `silinfo` object, as returned by `nk()`
k	a single value among the candidate `k` used in `nk()`

Details

Extract the fine-grained silhouette information of one `k` from an `nk()` result: one row per sample, with the cluster the sample was assigned to, its neighbour cluster and its silhouette width.

Value

a `data.frame` with one row per sample and the columns `sample`, `cluster`, `neighbor` and `sil_width`. The rows keep the order in which the clusterer returned them – for `cluster::pam()` that is by cluster and, within a cluster, by decreasing `sil_width` – so the rows should be aligned by `sample` rather than by position. The per-cluster average silhouette widths are attached as the `"clus.avg.widths"` attribute, and are also available as `x$silinfo[[i]]$clus.avg.widths`, where `i` is the position of `k` in `x$k`.

Author(s)

Guangchuang Yu

See Also

`nk()`, `autoplot()`

Examples

```
x <- nk(iris[,-5], 2:4)
head(silinfo_widths(x, 3))
attr(silinfo_widths(x, 3), "clus.avg.widths")
```

theme_dr *theme_dr*

Description

Dimensional reduction scatter plot axis theme

Usage

```
theme_dr(
  xlength = 0.3,
  ylength = 0.3,
  arrow = grid::arrow(length = unit(0.15, "inches"), type = "closed")
)
```

Arguments

xlength	length of x axis
ylength	length of y axis
arrow	arrow specification, as created by 'grid::arrow()'

Value

a theme object with shorten axes

Author(s)

Guangchuang Yu

theme_noaxis *theme_noaxis*

Description

theme that remove axis

Usage

```
theme_noaxis(...)
```

Arguments

...	additional theme setting

Value

a theme object that disable axes

Author(s)

Guangchuang Yu

Index

autoplot(), *9*
available_methods, 2
available_methods(), *5*

dr, 3
dr(), *4–6*
dr_compare, 4
dr_extract, 5

element_line2, 6

fortify(), *3, 4*, *6*

nk, 7
nk(), *9*

silinfo_widths, 9

theme_dr, 10
theme_noaxis, 10