Package: vizClust (via r-universe)

June 8, 2026

Type Package

Title Visualization and Exploration of Cluster Transitions

Version 0.1.0

Description Provides tools to explore and visualize transitions between clusters in multivariate data. The package generates pseudo-samples by interpolating between cluster medoids, enabling the study of gradual changes in feature space. It also computes k-nearest neighbors (KNN)-based statistics to relate pseudo-samples to real data and summarize variable behavior using mean, median, or standard deviation. Finally, the package offers interactive visualizations of variable trajectories along cluster transitions, including both direct trajectory plots and bootstrap-based interactive plots with confidence intervals to assess variability and uncertainty across the transition path.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

Imports ggplot2, ggiraph, reshape2, FNN, dplyr, tibble

Suggests webshot, cluster, webshot2, phenomap

NeedsCompilation no

Author Elsa Arribas [aut, cre], YingHong Chen [ctb], Ferran Reverter [ctb]

Maintainer Elsa Arribas <elsaarribasg@gmail.com>

Depends R (>= 4.1.0)

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

Date/Publication 2026-06-08 19:40:02 UTC

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

RemoteRef HEAD

RemoteSha 6045e46afa50a740043252029bd1346d03f4d6f6

Contents

`get_interval` *Interactive plot of bootstrap confidence intervals*

Description

Generates an interactive plot displaying the statistic value and its 95% bootstrap confidence intervals at each step of the transition between two clusters.

Usage

```
get_interval(data, nn_idx, B = 1000, vars = NULL, n_vars = NULL)
```

Arguments

`data`	A numeric data frame or matrix containing the original dataset, where rows represent samples and columns represent variables.
`nn_idx`	A matrix of nearest neighbor indices obtained from `knn_statistics()`. Each row represents a transition step and contains the indices of the k nearest neighbors associated with that step.
`B`	Number of bootstrap iterations used to estimate confidence intervals. Must be a positive integer. Default is 1000.
`vars`	Optional character vector specifying the variables to include in the plot. If provided, only the selected variables are displayed.
`n_vars`	Optional integer specifying the number of variables to display. The variables with the highest variance along the transition are selected. Ignored if `vars` is provided.

Details

The function performs the following steps:

1. Extracts the k-nearest neighbors for each step along the transition.
2. Computes bootstrap samples of the mean for each variable.
3. Estimates 95% confidence intervals using the bootstrap distribution.
4. Generates an interactive plot displaying the mean trajectories together with their confidence intervals.

Value

An interactive visualization displaying the trajectories of the selected variables across transition steps, together with bootstrap-based 95% confidence intervals and interactive tooltips containing variable names and interval values.

See Also

pseudosamples(), knn_statistics(), plot_explorer()

Examples

```
## Load example dataset
data(iris)

## Keep only numeric variables and scale
iris_scaled <- as.data.frame(scale(iris[, -5]))

## Perform PAM clustering
set.seed(123)
pam_iris <- cluster::pam(iris_scaled, k = 2)

## Extract medoids and generate pseudo-samples
medoids <- pam_iris$medoids
pseudo  <- pseudosamples(medoids, c1 = 1, c2 = 2, n_points = 20)

## Run KNN statistics
knn_res <- knn_statistics(iris_scaled, pseudo, k = 5, fun = "mean")

## Plot with bootstrap confidence intervals for all variables
get_interval(iris_scaled, knn_res$nn_idx, B = 100)

## Plot top 2 variables by variance
get_interval(iris_scaled, knn_res$nn_idx, B = 100,
             n_vars = 2)  ## Results for top variance variables

## Plot specific variables
get_interval(iris_scaled, knn_res$nn_idx, B = 100,
             vars = c("Sepal.Length", "Sepal.Width"))  ## Results for selected variables
```

---

knn_statistics *KNN-based summary statistics for pseudo-samples*

Description

This function maps pseudo-samples onto real data using k-nearest neighbors (KNN) and computes summary statistics for each variable, including the mean, median and standard deviation.

Usage

```
knn_statistics(data, pseudo.sample, k, fun = "mean")
```

Arguments

data	A numeric matrix or data frame containing the original dataset, where rows represent observations and columns represent variables.
pseudo.sample	A data frame containing pseudo-samples generated by pseudosamples().
k	Number of the nearest neighbors to consider.
fun	Character string specifying the summary statistic to compute for each variable. Supported values are "mean", "median", and "sd".

Details

For each pseudo-sample, the function identifies the k nearest neighbors in the original dataset and computes a summary statistic for each variable across the selected neighbors.

Supported summary statistics include:

- "mean": mean of the neighboring observations.
- "median": median of the neighboring observations.
- "sd": standard deviation of the neighboring observations.

Value

A list containing:

explorer	A data frame containing the summary statistics computed from the k nearest neighbors for each pseudo-sample and variable.
nn_idx	A matrix of nearest-neighbor indices, where each row corresponds to a pseudo-sample and each column to a neighboring observation. This object can be used as input for get_interval() to estimate bootstrap confidence intervals and produce interactive visualizations.

See Also

pseudosamples(), get_interval(), plot_explorer()

Examples

```
## Load example dataset
data(iris)

## Keep only numeric variables and scale
iris_scaled <- as.data.frame(scale(iris[, -5]))

## Perform PAM clustering
set.seed(123)
pam_iris <- cluster::pam(iris_scaled, k = 2)

## Extract medoids and generate pseudo-samples
medoids <- pam_iris$medoids
pseudo  <- pseudosamples(medoids, c1 = 1, c2 = 2, n_points = 20)
```

```
## Run KNN statistics with mean summary
knn_res <- knn_statistics(iris_scaled, pseudo, k = 15, fun = "mean")

head(knn_res$explorer)  ## Results for the explorer data frame
head(knn_res$nn_idx)    ## Results for the KNN indices
```

`plot_explorer` *Visualization of variable trajectories across cluster transitions*

Description

This function visualizes the evolution of selected variables along the pseudo-sample path using an interactive line plot.

Usage

```
plot_explorer(explorer, vars = NULL, n_vars = NULL)
```

Arguments

`explorer`	A data frame containing summarized values returned by `knn_statistics()`.
`vars`	Optional character vector specifying the variables to include in the plot.
`n_vars`	Optional integer indicating the number of variables to display. Variables are selected according to their variance across the transition.

Details

This function generates an interactive line plot showing how selected variables evolve along the transition path defined by the pseudo-samples.

The input `explorer` is typically obtained from `knn_statistics()`, where rows represent transition steps between clusters and columns represent variables.

Variable selection can be controlled as follows:

- If `vars` is provided, only the specified variables are displayed.
- If `n_vars` is provided, the variables with the highest variance across the transition are selected.
- If neither argument is provided, all variables are displayed.

The function reshapes the data into long format and creates an interactive visualization using `ggplot2` and `ggiraph`, allowing users to explore variable trajectories dynamically.

Value

An interactive `ggiraph` object representing a line plot of variable trajectories across the transition.

Each line corresponds to a variable, and each point along the x-axis represents a transition step between clusters.

See Also

pseudosamples(), knn_statistics(), get_interval()

Examples

```
## Load example dataset
data(iris)

## Keep only numeric variables and scale
iris_scaled <- as.data.frame(scale(iris[, -5]))

## Perform PAM clustering
set.seed(123)
pam_iris <- cluster::pam(iris_scaled, k = 2)

## Extract medoids and generate pseudo-samples
medoids <- pam_iris$medoids
pseudo  <- pseudosamples(medoids, c1 = 1, c2 = 2, n_points = 20)

## Run KNN statistics
knn_res <- knn_statistics(iris_scaled, pseudo, k = 15, fun = "mean")

## Plot all variables
plot_explorer(knn_res$explorer)

## Plot specific variables
plot_explorer(knn_res$explorer,
              vars = c("Sepal.Length", "Sepal.Width"))  ## Results for selected variables

## Plot top 2 variables by variance
plot_explorer(knn_res$explorer,
               n_vars = 2)  ## Results for top variance variables
```

`pseudosamples` *Pseudo-sample generation between cluster medoids*

Description

This function generates interpolated pseudo-samples along the linear transition between two cluster medoids. It is useful for exploring transitions between clusters in a multivariate feature space.

Usage

```
pseudosamples(medoids, c1, c2, n_points)
```

Arguments

medoids	A numeric matrix or data frame containing the cluster medoids, where rows represent clusters and columns represent variables.
c1	Index of the starting cluster.
c2	Index of the ending cluster.
n_points	Number of pseudo-samples to generate along the transition path between c1 and c2.

Details

The function computes a linear interpolation between two cluster medoids. A sequence of values for lambda between 0 and 1 is generated, and for each value, a new pseudo-sample is calculated as:

$$medoid\_c1 + \lambda * (medoid\_c2 - medoid\_c1)$$

This procedure produces a continuous trajectory in the feature space between the two clusters.

Value

A data frame with n_points rows and the same number of columns as medoids. Each row represents a pseudo-sample along the transition path between the two clusters.

See Also

knn_statistics(), plot_explorer(), get_interval()

Examples

```
## Load example dataset
data(iris)

## Keep only numeric variables and scale
iris_scaled <- scale(iris[, -5])

## Perform PAM clustering
set.seed(123)
pam_iris <- cluster::pam(iris_scaled, k = 2)

## Extract medoids
medoids <- pam_iris$medoids

## Generate pseudo-samples between cluster 1 and 2
pseudo <- pseudosamples(medoids, c1 = 1, c2 = 2, n_points = 20)
head(pseudo)
```

Index