Package 'sleepwalk'

February 26, 2020

Type Package

Title Interactively Explore Dimension-Reduced Embeddings

Version 0.3.0

Date 2020-02-26

Description A tool to interactively explore the
embeddings created by dimension reduction methods such as
Principal Components Analysis (PCA), Multidimensional Scaling (MDS),
T-distributed Stochastic Neighbour Embedding (t-SNE),
Uniform Manifold Approximation and Projection (UMAP) or any other.

License GPL-3

Imports jrc (>= 0.2.0), cowplot, httpuv, jsonlite, scales, ggplot2

RoxygenNote 7.0.2

URL https://anders-biostat.github.io/sleepwalk/

BugReports https://github.com/anders-biostat/sleepwalk/issues

NeedsCompilation no

Author Svetlana Ovchinnikova [aut, cre],
Simon Anders [aut]

Maintainer Svetlana Ovchinnikova <s.ovchinnikova@zmbh.uni-heidelberg.de>

Repository CRAN

Date/Publication 2020-02-26 17:50:02 UTC

R topics documented:

`.slw`	*Variable to store the information about the current* `sleepwalk` *session.*

Description

Variable to store the information about the current `sleepwalk` session.

Usage

```
.slw
```

Format

An object of class `environment` of length 0.

---

sleepwalk	*Interactively explore one or several 2D embeddings*

Description

A function to interactively explore a 2D embedding of some higher-dimensional point cloud, as produced by a dimension reduction method such as MDS, t-SNE, or the like.

Usage

```
sleepwalk(
  embeddings,
  featureMatrices = NULL,
  maxdists = NULL,
  pointSize = 1.5,
  titles = NULL,
  distances = NULL,
  same = c("objects", "features"),
  compare = c("embeddings", "distances"),
  saveToFile = NULL,
  ncol = NULL,
  nrow = NULL,
  on_selection = NULL,
  mode = c("canvas", "svg"),
  metric = "euclid",
  ...
)
```

Arguments

embeddings	either an $nx2$ embedding matrix (where n is a number of points) or a list of $n_i x2$ matrices - one for each embedding. If same = "objects" all embedding matrices must have the same number of rows.
featureMatrices	either an nxm matrix of point coordinates in the feature-dimension space or a list of such matrices - one for each embedding. The displayed distances will be calculated as Euclidean distances of the rows of these matrices. Alternatively, if same = "objects" it is possible to provide the distances directly via the distances argument. If same = "features" then all the points must be from the same feature space and therefore have the same number of columns. It is possible to use one feature matrix for all the embeddings.
maxdists	a vector of the maximum distances (in feature space) for each provided feature or distance matrix that should still be covered by the colour scale; higher distances are shown in light gray. This values can be changed later interactively. If not provided, maximum distances will be estimated automatically as median value of the distances.
pointSize	size of the points on the plots.
titles	a vector of titles for each embedding. Must be the same length as the list of embeddings.
distances	distances (in feature space) between points that should be displayed as colours. This is an alternative to featureMatrices if same = "objects".
same	defines what kind of distances to show; must be either "objects" or "features". Use same = "objects" when all the embeddings show the same set of points. In this case, each embedding is colored to show the distance of the selected point to all other points. The same or different features can be supplied as featureMatrices, to use the same or different distances in the different embeddings. same = "features" is used to compare different sets of points (e.g. samples from different patients, or different batches) in the same feature space. In this case the distance is calculated from the selected point to all other points (including those in other embeddings).
compare	defines what kind of comparison to perform; must be either "embeddings" or "distances". If compare == "embeddings", then in each of the displayed embeddings all the points will be coloured the same way, even if different feature or distance matrices are provided. This allows one to immediately identify the corresponding points and neighbourhoods in each of the embeddings. If commpare == "distances", point colours for each embedding are calculated independently. This allows, for instance, to compare different metrics or show an additional layer of information, when exploring an embedding. This parameter has no effect if same == "features".
saveToFile	path to the .html file where to save the plots. The resulting page will be fully interactive and contain all the data. If this is NULL, than the plots will be shown as the web page in your default browser. Note, that if you try to save that page, using your browser's functionality, it'll become static.
ncol	number of columns in the table, where all the embeddings are placed.

nrow	number of rows in the table, where all the embeddings are placed.
on_selection	a callback function that is called every time the user selects a group of points in the web browser. From the sleepwalk app it gets two arguments: The first one is a vector of indices of all the selected points and the second one is an index of an embedding from where the points were selected.
mode	defines whether to use Canvas or SVG to display points. Using Canvas is faster and allows to plot more points simultaneously, yet we currently consider SVG mode to be more stable and vigorously tested. In future versions SVG mode will be deprecated. Must be one of canvas or svg.
metric	specifies what metric to use to calculate distances from feature matrices. Currently only Euclidean ("euclid", default) and cosine ("cosine") are supported. This can be a single string (then the selected metric is used for all the charts) or a vector of strings - one per each chart.
...	Further arguments passed to openPage.

Details

The function opens a browser window and displays the embeddings as point clouds. When the user moves the mouse over a point, the point gets selected and all data points change colour such that their colour indicates the feature-space distance to the point under the mouse cursor. This allows to quickly and intuitively check how tight clusters are, how faithful the embedding is, and how similar the clusters are to each other.

Value

None.

Author(s)

Simon Anders, Svetlana Ovchinnikova

References

https://doi.org/10.1101/603589

Examples

```
#generate cockscrew-shaped 3D data with 3 additional noisy dimensions
ts <- c(rnorm(100), rnorm(200, 5), rnorm(150, 13), runif(200, min = -5, max = 20))

a <- 3
w <- 1

points <- cbind(30*cos(w * ts), 30*sin(w * ts), a * ts)

ndim <- 6
noise <- cbind(matrix(rnorm(length(ts) * 3, sd = 5), ncol = 3),
               matrix(rnorm(length(ts) * (ndim - 3), sd = 10), ncol = ndim - 3))
```

```
data <- noise
data[, 1:3] <- data[, 1:3] + points

pca <- prcomp(data)

#compare Euclidean distance with the real position on the helix
sleepwalk(list(pca$x[, 1:2], pca$x[, 1:2]), list(data, as.matrix(ts)),
          compare = "distances", pointSize = 3)
#the same, but with saving the web page to an HTML file
sleepwalk(list(pca$x[, 1:2], pca$x[, 1:2]), list(data, as.matrix(ts)),
          compare = "distances", pointSize = 3,
          saveToFile = paste0(tempdir(), "/test.html"))
```

---

slw_snapshot   *Make a snapshot of the currently running Sleepwalk app*

Description

This function produces a static plot that shows a given state of the currently active Sleepwalk app. Double click on a point in the web browser will generate a command that can reproduce this exact state of the app.

Usage

```
slw_snapshot(point, emb = 1, returnList = FALSE)
```

Arguments

point	an index of the focus point (i.e. the one, over which the mouse is hovering at the moment). To learn the index of a point double click on it in the web browser.
emb	an index of the embedding of the focus point. To learn the index of an embedding double click on any of its points in the web browser.
returnList	if TRUE returns a list of ggplot objects (one per embedding), that can be easily modified later. Otherwise returns a single ggplot objects with all the embeddings.

Value

a ggplot object or a list of ggplot objects.

Examples

```
data("iris")
sleepwalk(iris[, c(1,3)], iris[1:4], pointSize = 4)
slw_snapshot(10)
```

Index