Package 'sysAgNPs'

May 9, 2026

Title Systematic Quantification of AgNPs to Unleash their Potential for Applicability

Version 1.0.0

Description

There is variation across AgNPs due to differences in characterization techniques and testing metrics employed in studies. To address this problem, we have developed a systematic evaluation framework called 'sysAgNPs'. Within this framework, Distribution Entropy (DE) is utilized to measure the uncertainty of feature categories of AgNPs, Proclivity Entropy (PE) assesses the preference of these categories, and Combination Entropy (CE) quantifies the uncertainty of feature combinations of AgNPs. Additionally, a Markov chain model is employed to examine the relationships among the sub-features of AgNPs and to determine a Transition Score (TS) scoring standard that is based on steady-state probabilities. The 'sysAgNPs' framework provides metrics for evaluating AgNPs, which helps to unravel their complexity and facilitates effective comparisons among different AgNPs, thereby advancing the scientific research and application of these AgNPs.

License GPL (>= 3)

Language en-US

Encoding UTF-8

RoxygenNote 7.3.2

LazyData true

URL https://github.com/xitingwang-ida/sysAgNPs

Imports dplyr, expm, ggplot2, ggpubr, magrittr, patchwork, purrr, rio, tibble, tidyr, rlang, RColorBrewer, forcats, stats

NeedsCompilation no

Author Xiting Wang [aut, cre] (ORCID: <https://orcid.org/0009-0009-5235-0006>), Longfei Mao [aut, cph] (ORCID: <https://orcid.org/0000-0003-0759-0501>), Jiamin Hu [ctb] (ORCID: <https://orcid.org/0000-0003-3030-2117>)

Maintainer Xiting Wang <XitingWang2023@outlook.com>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2025-01-20 16:20:02 UTC

Contents

`binary_dataset` *A binary dataframe of datasets used to establish evaluation criteria.*

Description

A binary dataframe of datasets used to establish evaluation criteria.

Usage

```
data(binary_dataset)
```

Format

A dataframe with 600 rows and 50 variables.

"Plant parts or microbial sites" to "Anti-inflammatory" Subfeature of nano silver.

dataset *Nanosilver data set.*

Description

Nanosilver data set.

Usage

```
data(dataset)
```

Format

A dataframe with 600 rows and 15 variables.

"Synthesis methods" to "Applications" Features of nano silver.

sysAgNPs_score *sysAgNPs package application results in four evaluation scores.*

Description

sysAgNPs package application results in four evaluation scores.

Usage

```
data(sysAgNPs_score)
```

Format

A dataframe with 600 rows and 4 variables.

DE to TS Four evaluation scores.

`sys_CalculateAxisPath` *Calculate Axis Path This function is derived from the 'ggradar' package. https://github.com/ricardo-bion/ggradar/. Calculates x-y coordinates for a set of radial axes (one per variable being plotted in radar plot)*

Description

Calculate Axis Path This function is derived from the 'ggradar' package. https://github.com/ricardo-bion/ggradar/. Calculates x-y coordinates for a set of radial axes (one per variable being plotted in radar plot)

Usage

```
sys_CalculateAxisPath(var.names, min, max)
```

Arguments

`var.names`	list of variables to be plotted on radar plot
`min`	MININUM value required for the plotted axes (same value will be applied to all axes)
`max`	MAXIMUM value required for the plotted axes (same value will be applied to all axes)

Value

a dataframe of the calculated axis paths

`sys_CalculateGroupPath` *Calculate Group Path This function is derived from the 'ggradar' package. https://github.com/ricardo-bion/ggradar/. Converts variable values into a set of radial x-y coordinates*

Description

Calculate Group Path This function is derived from the 'ggradar' package. https://github.com/ricardo-bion/ggradar/. Converts variable values into a set of radial x-y coordinates

Usage

```
sys_CalculateGroupPath(df)
```

Arguments

df	a dataframe with Col 1 is group ('unique' cluster / group ID of entity) and Col 2-n are v1.value to vn.value - values (e.g. group/cluser mean or median) of variables v1 to v.n

Value

a dataframe of the calculated axis paths

Source

Code adapted from a solution posted by Tony M to https://stackoverflow.com/questions/9614433/creating-radar-chart-a-k-a-star-plot-spider-plot-using-ggplot2-in-r/.

---

sys_CE *Calculate the Combination Entropy*

Description

Calculate the average value of the probability of surprising level of the presence and absence of a particular category within the specific category to measure the average uncertainty of feature categories.

Usage

```
sys_CE(data, dataset)
```

Arguments

data	A dataframe that contains experimental data.
dataset	The dataset used to to calculate the ratio of the number of reporting a certain feature in the AgNPs dataset to the total number of samples.

Value

A dataframe including: 1. the ratio of the number of reporting a certain feature in the AgNPs dataset to the total number of samples; 2. pc:the probability of the feature combination occurring; 3. Hi:the probability of surprising level of the presence and absence of feature combinations to measure the uncertainty of feature combination.

Examples

```
data(dataset)
users_data <- dataset
CE <- sys_CE(users_data, dataset)
```

sys_DE *Calculate the Distribution Entropy*

Description

Measure the distribution variability of the presence and absence of feature categories.

Usage

```
sys_DE(data)
```

Arguments

data	A dataframe that contains experimental data.

Value

A dataframe including 1. the number of feature in a certain category; 2. the total number of features in the sample; 3. the average value to measure the average uncertainty of feature categories

Examples

```
data(dataset)
users_data <- dataset
DE = sys_DE(users_data)
```

sys_discretize *Convert categorical variables into discrete variables*

Description

Convert categorical variables into discrete variables

Usage

```
sys_discretize(dataset, vars_to_discretize = NULL)
```

Arguments

dataset	A dataframe of dataset. Datasets used to establish evaluation criteria.
vars_to_discretize	Variables or columns to be discretized. Default is NULL.

Value

A binary dataframe.

Examples

```
data(dataset)
dis_data <- sys_discretize(dataset, c("Shape", "pH"))
```

`sys_eval_cri` *Build Transition Scores criteria*

Description

This function evaluates the criteria for a binary dataset by calculating the transfer probability matrix and iterating to obtain the transfer probability vector.

Usage

```
sys_eval_cri(binary_dataset, n_iter, vars_to_discretize = NULL)
```

Arguments

`binary_dataset`	A binary dataframe of datasets used to establish evaluation criteria.
`n_iter`	The number of iterations to reach the steady state.
`vars_to_discretize`	Variables or columns to be discretized. Default is NULL.

Value

A dataframe containing the scores of nanomaterial features.

Examples

```
data(dataset)
binary_dataset <- dataset
var_dis <- c("Synthesis methods", "pH", "Temperature (℃)",
"Zeta potential (mV)","Size (nm)", "Shape", "Applications")
criteria <- sys_eval_cri(binary_dataset, 6, var_dis)
```

sys_funcCircleCoords *Generate circle coordinates This function is derived from the 'ggradar' package. https://github.com/ricardo-bion/ggradar/. Generate coordinates to draw a circle.*

Description

Generate circle coordinates This function is derived from the 'ggradar' package. https://github.com/ricardo-bion/ggradar/. Generate coordinates to draw a circle.

Usage

```
sys_funcCircleCoords(center = c(0, 0), r = 1, npoints = 100)
```

Arguments

center	coordinate for centroid
r	radius
npoints	number of coordinates to generate

Value

a dataframe

Source

Adapted from Joran's response to https://stackoverflow.com/questions/6862742/draw-a-circle-with-ggplot2/.

sys_generate_color_values *Generate Dynamic Color Values This function is derived from the 'ggradar' package. https://github.com/ricardo-bion/ggradar/. This function dynamically generates a vector of color values based on the number of groups. It uses RColorBrewer for smaller sets of groups and generates a gradient for larger sets.*

Description

Generate Dynamic Color Values This function is derived from the 'ggradar' package. https://github.com/ricardo-bion/ggradar/. This function dynamically generates a vector of color values based on the number of groups. It uses RColorBrewer for smaller sets of groups and generates a gradient for larger sets.

Usage

```
sys_generate_color_values(num_groups)
```

Arguments

num_groups	The number of groups for which to generate color values.

Value

A character vector of color values.

Examples

```
sys_generate_color_values(5)
sys_generate_color_values(20)
```

---

sys_ggradar *This function is derived from the 'ggradar' package.* https://github.com/ricardo-bion/ggradar/.

Description

This function is derived from the 'ggradar' package. https://github.com/ricardo-bion/ggradar/.

Usage

```
sys_ggradar(
  plot.data,
  base.size = 15,
  font.radar = "sans",
  values.radar = c("0%", "50%", "100%"),
  axis.labels = colnames(plot.data)[-1],
  grid.min = 0,
  grid.mid = 0.5,
  grid.max = 1,
  centre.y = grid.min - ((1/9) * (grid.max - grid.min)),
  plot.extent.x.sf = 1,
  plot.extent.y.sf = 1.2,
  x.centre.range = 0.02 * (grid.max - centre.y),
  label.centre.y = FALSE,
  grid.line.width = 0.5,
  gridline.min.linetype = "longdash",
  gridline.mid.linetype = "longdash",
  gridline.max.linetype = "longdash",
  gridline.min.colour = "grey",
  gridline.mid.colour = "#007A87",
  gridline.max.colour = "grey",
  grid.label.size = 6,
  gridline.label.offset = -0.1 * (grid.max - centre.y),
  label.gridline.min = TRUE,
  label.gridline.mid = TRUE,
```

```
  label.gridline.max = TRUE,
  axis.label.offset = 1.15,
  axis.label.size = 5,
  axis.line.colour = "grey",
  group.line.width = 1.5,
  group.point.size = 6,
  group.colours = NULL,
  background.circle.colour = "#D7D6D1",
  background.circle.transparency = 0.2,
  plot.legend = if (nrow(plot.data) > 1) TRUE else FALSE,
  legend.title = "",
  plot.title = "",
  legend.text.size = 14,
  legend.position = "left",
  fill = FALSE,
  fill.alpha = 0.5,
  draw.points = TRUE,
  point.alpha = 1,
  line.alpha = 1
)
```

Arguments

plot.data	dataframe comprising one row per group
base.size	text size
font.radar	text font family
values.radar	values to print at minimum, 'average', and maximum gridlines
axis.labels	names of axis labels if other than column names supplied via plot.data
grid.min	value at which mininum grid line is plotted
grid.mid	value at which 'average' grid line is plotted
grid.max	value at which maximum grid line is plotted
centre.y	value of y at centre of plot
plot.extent.x.sf	controls relative size of plot horizontally
plot.extent.y.sf	controls relative size of plot vertically
x.centre.range	controls axis label alignment
label.centre.y	whether value of y at centre of plot should be labelled
grid.line.width	width of gridline
gridline.min.linetype	line type of minimum gridline
gridline.mid.linetype	line type of 'average' gridline

`gridline.max.linetype`
: line type of maximum gridline

`gridline.min.colour`
: colour of minimum gridline

`gridline.mid.colour`
: colour of 'average' gridline

`gridline.max.colour`
: colour of maximum gridline

`grid.label.size`
: text size of gridline label

`gridline.label.offset`
: displacement to left/right of central vertical axis

`label.gridline.min`
: whether or not to label the mininum gridline

`label.gridline.mid`
: whether or not to label the 'mininum'average' gridline

`label.gridline.max`
: whether or not to label the maximum gridline

`axis.label.offset`
: vertical displacement of axis labels from maximum grid line, measured relative to circle diameter

`axis.label.size`
: text size of axis label

`axis.line.colour`
: colour of axis line

`group.line.width`
: line width of group

`group.point.size`
: point size of group

`group.colours` colour of group

`background.circle.colour`
: colour of background circle/radar

`background.circle.transparency`
: transparency of background circle/radar

`plot.legend` whether to include a plot legend

`legend.title` title of legend

`plot.title` title of radar plot

`legend.text.size`
: text size in legend

`legend.position`
: position of legend, valid values are "top", "right", "bottom", "left"

`fill` whether to fill polygons

`fill.alpha` if filling polygons, transparency values

`draw.points` whether to draw points

`point.alpha` alpha for points, can be a single value or vector

`line.alpha` alpha for lines, can be a single value or vector

Value

a ggplot object

Source

Most of the code is from http://rstudio-pubs-static.s3.amazonaws.com/5795_e6e6411731bb4f1b9cc7eb49499c208
html.

`sys_iter` *Obtain the transition probability of each iteration*

Description

Loop "n_iter" times to obtain the transition probability of each iteration.

Usage

```
sys_iter(binary_dataset, n_iter, vars_to_discretize = NULL)
```

Arguments

`binary_dataset`	A binary dataframe of datasets used to establish evaluation criteria.
`n_iter`	The number of iterations to reach the steady state.
`vars_to_discretize`	Variables or columns to be discretized. Default id NULL.

Value

A dataframe containing the number of iterations and the transition probability of each iteration.

Examples

```
data(dataset)
iter_prob <- sys_iter(dataset, 6, c("Shape", "pH"))
```

sys_line_radar *Line and Radar Plot of sysAgNPs score*

Description

Line and Radar Plot of sysAgNPs score

Usage

```
sys_line_radar(sysAgNPs_score, num_plots)
```

Arguments

sysAgNPs_score	A dataframe containing four columns of numeric data.
num_plots	The range of the graph to be output and saved can be a vector or a single value.

Value

A ggplot object.

sys_PE *Calculate the Proclivity Entropy*

Description

Measure the preference of feature categories.

Usage

```
sys_PE(data)
```

Arguments

data	A dataframe that contains experimental data.

Value

A dataframe including 1. the number of feature in a certain category; 2. the total number of features in the sample; 3. the expected value to measure the average description level across different feature categories.

Examples

```
data(dataset)
users_data <- dataset
PE = sys_PE(users_data)
```

sys_steady *Iterate to obtain the steady state probability*

Description

Change the values of the constraints step by step and record the number of iterations to reach the steady state.

Usage

```
sys_steady(
  binary_dataset,
  tran_matrix,
  tol_vec = c(0.01, 0.001, 1e-04, 1e-05, 1e-06, 1e-07)
)
```

Arguments

binary_dataset	A binary dataframe of datasets used to establish evaluation criteria.
tran_matrix	A transfer probability matrix.
tol_vec	A smaller constants used as constraints.

Value

A data frame containing the constraints and the number of iterations to reach the steady state.

Examples

```
data(binary_dataset)
data(tran_matrix)
tol_iter <- sys_steady(binary_dataset, tran_matrix, 1e-5)
```

sys_tran *Calculate transition probability matrix*

Description

Calculate transition probability matrix

Usage

```
sys_tran(binary_dataset)
```

Arguments

binary_dataset	A binary dataframe of datasets used to establish Transition Scores criteria.

Value

A transfer probability matrix.

Examples

```
data(binary_dataset)
tran_matrix <- sys_tran(binary_dataset)
```

`sys_TS` *Calculate the Transition Scores*

Description

Calculate the Transition Scores

Usage

```
sys_TS(data, dataset, n_iter, vars_to_discretize)
```

Arguments

`data`	A dataframe that contains experimental data.
`dataset`	A binary dataframe. Datasets used to establish evaluation criteria.
`n_iter`	The number of iterations to reach the steady state.
`vars_to_discretize`	Variables or columns to be discretized. Default is NULL.

Value

A dataframe that contains sysAgNPs scores.

`tran_matrix` *A transfer probability matrix.*

Description

A transfer probability matrix.

Usage

```
data(tran_matrix)
```

Format

A matrix with 50 rows and 50 columns.

V1 to V50 Subfeature of nano silver.

Index