Package ‘wilson’

November 18, 2019

Type Package

Title Web-Based Interactive Omics Visualization

Version 2.3.1

Description Tool-set of modules for creating web-based applications that use plot based strategies to visualize and analyze multi-omics data. This package utilizes the ‘shiny’ and ‘plotly’ frameworks to provide a user friendly dashboard for interactive plotting.

URL https://github.com/loosolab/wilson/

BugReports https://github.com/loosolab/wilson/issues/

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports shiny, data.table, ggplot2, plotly (> 4.8.0), scales, shinydashboard, DT (>= 0.3), colourpicker, RColorBrewer, shinyjs, viridis, rje, grDevices, grid, plyr, circlize, ComplexHeatmap, stats, gplots, reshape, rintrojs, RJSONIO, ggrepel (>= 0.6.12), DESeq2, rjson, FactoMineR, factoextra, heatmaply (>= 0.14.1), shinyCSSloaders, log4r, openssl, methods, R6, magrittr, zip

RoxygenNote 6.1.1

biocViews

Suggests knitr, rmarkdown, testthat, vdiffr, stringi

VignetteBuilder knitr

NeedsCompilation no

Author Hendrik Schultheis [aut, cre], Jens Preussner [aut], Looso Mario [aut]

Maintainer Hendrik Schultheis <hendrik.schultheis@mpi-bn.mpg.de>

Repository CRAN

Date/Publication 2019-11-18 14:10:02 UTC
R topics documented:

and ......................................................... 3
andUI ....................................................... 4
categoricalPalettes ........................................ 4
Clarion ..................................................... 5
colorPicker ............................................... 6
colorPickerUI ............................................. 7
columnSelector .......................................... 7
columnSelectorUI ........................................ 7
create_geneview ......................................... 8
create_heatmap .......................................... 10
create_pca ............................................... 11
create_scatterplot ...................................... 12
divergingPalettes ....................................... 13
download .................................................. 14
equalize ................................................... 14
featureSelector ......................................... 15
featureSelectorGuide .................................... 16
featureSelectorUI ....................................... 16
forceArgs ............................................... 17
geneView ............................................... 17
geneViewGuide ......................................... 18
geneViewUI ............................................. 19
global_cor_heatmap ..................................... 19
global_cor_heatmapUI ................................... 20
global_cor_heatmap_guide ............................... 20
heatmap .................................................. 21
heatmapGuide ............................................ 21
heatmapUI ............................................... 22
label ...................................................... 22
labelUI ................................................... 23
limit ...................................................... 24
limitUI ................................................... 24
log_message ............................................. 25
marker .................................................... 25
markerUI .................................................. 26
orNumeric ................................................ 26
orNumericUI ............................................. 27
orTextual ............................................... 28
orTextualUI ............................................. 29
parser .................................................... 29
parse_MaxQuant ......................................... 30
pca ...................................................... 31
pcaGuide ............................................... 31
pcaUI .................................................... 32
release_questions ....................................... 32
scatterPlot .............................................. 33
Description

This function evaluates output from multiple OR modules by combining with a logical and.

Usage

```r
and(input, output, session, data, show.elements = NULL,
   element.grouping = NULL, column.labels = NULL, delimiter = NULL,
   multiple = TRUE, contains = FALSE, ranged = FALSE, step = 100,
   reset = NULL)
```

Arguments

- **input**: Shiny’s input object.
- **output**: Shiny’s output object.
- **session**: Shiny’s session object.
- **data**: The input data.frame for which selection should be provided. Evaluates an OR module for each column (Supports reactive).
- **show.elements**: A Vector of column names determining which OR modules are shown. Defaults to names(data). (Supports reactive)
- **element.grouping**: Group features in boxes. (Data.table: first column = columnnames, second column = groupnames) (Supports reactive)
- **column.labels**: Additional labels for the columns, defaults to names(data).
- **delimiter**: A single character, or a vector indicating how column values are delimited. (Fills vector sequentially if needed) (Supports reactive)
- **multiple**: Whether or not textual ORs should allow multiple selections. (Fills vector sequentially if needed) (Supports reactive)
- **contains**: Whether or not textual ORs are initialized as textInput checking entries for given string. (Fills vector sequentially if needed) (Supports reactive)
categoricalPalettes

ranged Whether or not numeric ORs are ranged. (Fills vector sequentially if needed)(Supports reactive)
step Set numeric ORs slider steps. (Fills vector sequentially if needed)(Supports reactive)
reset Reactive which will cause a UI reset on change.

Value

A reactive containing named list with a boolean vector of length nrow(data) (bool), indicating whether an observation is selected or not and a vector of Strings showing the used filter (text).

andUI

AND module UI representation

Description

The AND module connects filtering and selection across multiple columns of a data.frame. Columns of class boolean, character or factor will be represented as textual ORs, numeric columns as numerical OR.

Usage

andUI(id)

Arguments

id The ID of the modules namespace.

Value

A list with HTML tags from tag.

categoricalPalettes

Function to generate categorical (qualitative) color palettes

Description

Function to generate categorical (qualitative) color palettes

Usage

categoricalPalettes(n)

Arguments

n Number of colors to generate
Value
A data.table with (named) color palettes of length n

Clarion R6-class definition

Description
Use this to create a clarion object. This object is used by all top-level wilson modules.

Arguments

- **header**: A named list. Defaults to NULL.
- **metadata**: Clarion metadata in form of a data.table.
- **data**: Data.table according to metadata.
- **validate**: Logical value to validate on initialization. Defaults to TRUE.

Fields
- **header**: List of global information regarding the whole experiment.
- **metadata**: Data.table with additional information for each column.
- **data**: Data.table containing experiment result data.

Methods

- **get_id()**: Returns name of unique identifier column. Assumes first feature to be unique if not specified.
- **get_name()**: Returns name of name column. If not specified return unique Id.
- **get_delimiter()**: Return delimiter used within multi-value fields (no delimiter = NULL).
- **is_delimited(x)**: Logical whether the given column name is delimited.
- **get_factors()**: Returns a data.table columns: key and factor(s) if any. Named factors (e.g. factor1="name") will be cropped to their name.
- **validate(solve = TRUE)**: Check the object for inconsistencies. For solve = TRUE try to resolve some warnings.

Examples

```r
## Not run:
# initializing a new object
object <- Clarion$new(header, metadata, data, validate = TRUE)

# create a deep copy
object_copy <- object$clone(deep = TRUE)

## End(Not run)
```
**Description**

Provides server logic for the colorPicker2 module.

**Usage**

```r
colorPicker(input, output, session, num.colors = 256, 
  distribution = "all", winsorize = NULL, selected = NULL)
```

**Arguments**

- `input`  Shiny’s input object
- `output` Shiny’s output object
- `session` Shiny’s session object
- `num.colors` Define length of colorpalette vector (Default = 256).
- `distribution` Decide which palettes are selectable. One or more of list("sequential", "diverging", "categorical"). Defaults to “all” (Supports reactive).
- `winsorize` Numeric vector of two. Dynamically change lower and upper limit (supports reactive). Defaults to NULL.
- `selected` Set the default selected palette.

**Details**

A custom colorpalette’s return will be NULL if there is something wrong with it.

equalize will be returned as FALSE if not selected.

**Value**

Reactive containing list(palette = c(colors), name = palette_name, transparency = Integer, reverse = Boolean, winsorize = NULL or a two-component vector containing lower and upper limits).
**colorPickerUI**  

**colorPicker module UI representation**

**Description**

The function creates HTML tag definitions of its representation based on the parameters supplied. Currently, two UI can be created for the user to choose either (a) colors from a given color scheme, or (b) choose one or more single colors.

**Usage**

```r
colorPickerUI(id, label = "Color scheme", custom = FALSE, 
multiple = FALSE, show.reverse = TRUE, show.scaleoptions = TRUE, 
show.transparency = TRUE)
```

**Arguments**

- **id**  
  The ID of the modules namespace.
- **label**  
  Either a character vector of length one with the label for the color scheme dropdown, or a character vector containing labels of the single colors.
- **custom**  
  Boolean if TRUE custom colors can be selected (Default = FALSE).
- **multiple**  
  Boolean value, if set to TRUE custom color palettes can be made. Only if custom = TRUE (Default = FALSE).
- **show.reverse**  
  Logical, whether or not to show the reverse switch (Default = TRUE).
- **show.scaleoptions**  
  Logical, whether or not to show color scaling option winsorize (Default = TRUE).
- **show.transparency**  
  Logical, whether or not to show the transparency slider (Default = TRUE).

**Value**

A list with HTML tags from `tag`.

---

**columnSelector**  

**columnSelector module server logic**

**Description**

columnSelector module server logic

**Usage**

```r
columnSelector(input, output, session, type.columns, type = NULL, 
column.type.label = "Type of Column", label.label = "Label", 
multiple = TRUE, none = FALSE, sep = "", suffix = NULL)
```
Arguments

- **input**: Shiny's input object
- **output**: Shiny's output object
- **session**: Shiny's session object
- **type.columns**: data.table: (Supports reactive) key = columnnames (id) level = datalevel/ type of column label = optional, used instead of id sub_label = optional, added to id/ label
- **type**: The type (contrast/group/sample of the type dropdown menu, selected in step 1 (upper dropdown). Defaults to unique(type.columns[,2]) (Supports reactive)
- **column.type.label**: Changes the label of the first UI element
- **label.label**: Change label above label text input.
- **multiple**: Boolean value whether multiple values can be selected in second selector. (Default = TRUE)
- **none**: If TRUE adds "None to secondSelector and select is. (Default = FALSE)
- **sep**: Used to separate labels on multi value selection.
- **suffix**: Added to label only on multiple = FALSE (supports reactive). Also uses sep as separator.

Value

Returns the input. As named list: names("type", "selected.columns", "label")

columnSelectorUI

columnSelector module UI representation

Description

columnSelector module UI representation

Usage

columnSelectorUI(id, label = FALSE, title = NULL)

Arguments

- **id**: The ID of the modules namespace.
- **label**: Boolean value; if true include a text input field with the desired axis label (this should be preset with the headline of the column)
- **title**: String which is displayed as module title. (Default = NULL)

Value

A list from tag with the UI elements.
Method for geneView creation

Usage

create_geneview(data, grouping, plot.type = "line",
      facet.target = "gene", facet.cols = 2, colors = NULL,
      ylabel = NULL, ylim = NULL, gene.label = NULL,
      plot.method = "static", width = "auto", height = "auto",
      ppi = 72, scale = 1)

Arguments

data data.table containing plot data

grouping data.table metadata containing: column1 : key column2 : factor1

plot.type String specifying which plot type is used c("box", "line", "violin", "bar").

facet.target Target to plot on x-Axis c("gene", "condition").

facet.cols Number of plots per row.

colors Vector of colors used for color palette

ylabel Label of the y-axis (default = NULL).

ylimits Vector defining scale of y-axis (default = NULL).

gene.label Vector of labels used instead of gene names (default = NULL).

plot.method Choose which method used for plotting. Either "static" or "interactive" (Default = "static").

width Set the width of the plot in cm (default = "auto").

height Set the height of the plot in cm (default = "auto").

ppi Pixel per inch (default = 72).

scale Modify plot size while preserving aspect ratio (Default = 1).

Details

Width/ height limit = 500. If exceeded default to 500 and issue exceed_size = TRUE.

Value

Returns depending on plot.method list(plot = ggplot/ plotly object, width = width in cm, height = height in cm, ppi = pixel per inch, exceed_size = Boolean).
create_heatmap  

Method for heatmap creation

Description

Method for heatmap creation

Usage

create_heatmap(data, unitlabel = "auto", row.label = TRUE,
row.custom.label = NULL, column.label = TRUE,
column.custom.label = NULL, clustering = "none",
clustdist = "auto", clustmethod = "auto", colors = NULL,
winsorize.colors = NULL, plot.method = "static", width = "auto",
height = "auto", ppi = 72, scale = 1)

Arguments

data  data.table containing plot data. First column contains row labels.
unitlabel  label of the colorbar
row.label  Logical whether or not to show row labels.
row.custom.label  Vector of custom row labels.
column.label  Logical whether or not to show column labels.
column.custom.label  Vector of custom column labels.
clustering  How to apply clustering on data. c("none", "both", "column", "row")
clustdist  Which cluster distance to use. See heatmapr.
clustmethod  Which cluster method to use. See heatmapr.
colors  Vector of colors used for color palette.
winsorize.colors  NULL or a vector of length two, giving the values of colorbar ends (default = NULL).
plot.method  Choose which method is used for plotting. Either "plotly" or "complexHeatmap" (Default = "complexHeatmap").
width  Set width of plot in cm (Default = "auto").
height  Set height of plot in cm (Default = "auto").
ppi  Pixel per inch (default = 72).
scale  Modify plot size while preserving aspect ratio (Default = 1).

Details

Width/ height limit = 500. If exceeded default to 500 and issue exceed_size = TRUE.
create_pca

Value

Returns list(plot = complexHeatmap/plotly object, width = width in cm, height = height in cm, ppi = pixel per inch, exceed_size = Boolean whether width/height exceeded max) depending on plot.method.

Description

Method for pca creation.

Usage

create_pca(data, color.group = NULL, color.title = NULL, palette = NULL, shape.group = NULL, shape.title = NULL, shapes = c(15:25), dimension.a = 1, dimension.b = 2, dimensions = 6, on.columns = TRUE, labels = FALSE, custom.labels = NULL, pointsize = 2, labelsize = 3, width = 28, height = 28, ppi = 72, scale = 1)

Arguments

data data.table from which the plot is created (First column will be handled as row-names if not numeric).

color.group Vector of groups according to samples (= column names).
color.title Title of the color legend.
palette Vector of colors used for color palette.
shape.group Vector of groups according to samples (= column names).
shape.title Title of the shape legend.
shapes Vector of shapes see points. Will recycle/cut off shapes if needed. Default = c(15:25)
dimension.a Number of dimension displayed on X-Axis.
dimension.b Number of dimension displayed on Y-Axis.
dimensions Number of dimensions to create.
on.columns Boolean perform pca on columns or rows.
labels Boolean show labels.
custom.labels Vector of custom labels. Will replace columnnames.
pointsize Size of the data points.
labelsize Size of texts inside plot (default = 3).
width Set the width of the plot in cm (default = 28).
height Set the height of the plot in cm (default = 28).
ppi Pixel per inch (default = 72).
scale Modify plot size while preserving aspect ratio (Default = 1).
create_scatterplot

Details

If width and height are the same axis ratio will be set to one (quadratic plot).
Width/ height limit = 500. If exceeded default to 500 and issue exceed_size = TRUE.

Value

A named list(plot = ggplot object, data = pca.data, width = width of plot (cm), height = height of plot (cm), ppi = pixel per inch, exceed_size = Boolean whether width/ height exceeded max).

data.table containing plot data column 1: id column 2, 3(, 4): x, y(, z)
data.labels Vector of labels used for data. Length has to be equal to nrow(data).
data.hovertext Character vector with additional hovertext. Length has to be equal to nrow(data).
transparency Set point transparency. See geom_point.
pointsize Set point size. See geom_point.
labelsize Set label size. See geom_text.
color Vector of colors used for color palette.
x_label Label x-Axis
y_label Label Y-Axis
z_label Label Z-Axis
density Boolean value, perform 2d density estimate.
line Boolean value, add reference line.
categorized Z-Axis (if exists) as categories.
**divergingPalettes**

- `highlight.data`: data.table containing data to highlight. Same structure as data.
- `highlight.labels`: Vector of labels used for highlighted data. Length has to be equal to `nrow(highlight.data)`.
- `highlight.hovertext`: Character vector with additional hovertext. Length has to be equal to `nrow(highlight.data)`.
- `highlight.color`: String with hexadecimal color-code.
- `xlim`: Numeric vector of two setting min and max limit of x-axis. See `lims`.
- `ylim`: Numeric vector of two setting min and max limit of y-axis. See `lims`.
- `colorbar.limits`: Vector with min, max values for colorbar (Default = NULL).
- `width`: Set plot width in cm (Default = "auto").
- `height`: Set plot height in cm (Default = "auto").
- `ppi`: Pixel per inch (default = 72).
- `plot.method`: Whether the plot should be 'interactive' or 'static' (Default = 'static').
- `scale`: Modify plot size while preserving aspect ratio (Default = 1).

**Details**

Width/ height limit = 500. If exceeded default to 500 and issue exceed_size = TRUE.

**Value**

Returns list(plot = ggplotly/ ggplot, width, height, ppi, exceed_size).

---

**divergingPalettes**  
*Function to generate diverging (two-sided) color palettes (e.g. for log2fc, zscore)*

**Description**

Function to generate diverging (two-sided) color palettes (e.g. for log2fc, zscore)

**Usage**

`divergingPalettes(n)`

**Arguments**

- `n`: Number of colors to generate

**Value**

A data.table with (named) color palettes of length n
### Download

**Function used for downloading.** Creates a zip container containing plot in png, pdf and user input in json format. Use inside `downloadHandler` content function.

**Usage**

```r
download(file, filename, plot, width, height, ppi = 72, 
save_plot = TRUE, ui = NULL)
```

**Arguments**

- `file` See `downloadHandler` content parameter.
- `filename` See `downloadHandler`.
- `plot` Plot to download.
- `width` in centimeter.
- `height` in centimeter.
- `ppi` pixel per inch. Defaults to 72.
- `save_plot` Logical if plot object should be saved as .RData.
- `ui` List of user inputs. Will be converted to JavaScript Object Notation. See `toJSON`.

**Value**

Path to zip archive invisibly. See `zipr`.

### Equalize

**Method to get equalized min/max values from vector**

**Description**

Method to get equalized min/max values from vector.

**Usage**

```r
equalize(values)
```

**Arguments**

- `values` Numeric vector or table
**featureSelector**

**Value**

Vector with c(min, max).

---

**featureSelector**

**featureSelector module server logic**

**Description**

featureSelector module server logic

**Usage**

```r
featureSelector(input, output, session, clarion, multiple = TRUE,
contains = FALSE, ranged = TRUE, step = 100, truncate = 30,
selection.default = "all")
```

**Arguments**

- `input` : Shiny’s input object.
- `output` : Shiny’s output object.
- `session` : Shiny’s session object.
- `clarion` : A clarion object. See Clarion. (Supports reactive)
- `multiple` : Whether or not textual ORs should allow multiple selections. (Fills vector sequentially if needed)(Supports reactive)
- `contains` : Whether or not textual ORs are initialized as textInput checking entries for given string. (Fills vector sequentially if needed)(Supports reactive)
- `ranged` : Whether or not numeric ORs are ranged. (Fills vector sequentially if needed)(Supports reactive)
- `step` : Set numeric ORs number of slider steps. (Fills vector sequentially if needed)(Supports reactive)
- `truncate` : Truncate datatable entries at x characters (Default = 30).
- `selection.default` : Decide whether everything or nothing is selected on default (no filters applied). Either "all" or "none" (Default = "all").

**Details**

Keep in mind that the order of features (columns in clarion$data) is the order in which multiple, contains, ranged and step are evaluated.

**Value**

Reactive containing names list: Selected data as reactive containing clarion object (object). Used filter to select data (filter).
featureSelectorGuide  
*featureSelector module guide*

**Description**

featureSelector module guide

**Usage**

featureSelectorGuide(session)

**Arguments**

- **session**  
  The shiny session

**Value**

A shiny reactive that contains the texts for the guide steps.

---

featureSelectorUI  
*featureSelector module UI representation*

**Description**

featureSelector module UI representation

**Usage**

featureSelectorUI(id)

**Arguments**

- **id**  
  The ID of the modules namespace

**Value**

A list with HTML tags from *tag*
**forceArgs**

*Force evaluation of the parent function's arguments.*

**Description**

Force evaluation of the parent function’s arguments.

**Usage**

```r
forceArgs(args)
```

**Arguments**

- `args` List of Argument names to force evaluation. Defaults to all named arguments see `match.call`.

**Details**

Similar to `forceAndCall` but used from within the respective function.

This method is not using `force` as it is restricted to it’s calling environment. Instead `get` is used.

---

**geneView**

*geneView's module server logic*

**Description**

Provides server logic for the geneView module.

**Usage**

```r
geneView(input, output, session, clarion, plot.method = "static", label.sep = ", ", width = "auto", height = "auto", ppi = 72, scale = 1)
```

**Arguments**

- `input` Shiny’s input object.
- `output` Shiny’s output object.
- `session` Shiny’s session object.
- `clarion` A clarion object. See `Clarion`. (Supports reactive)
- `plot.method` Choose which method is used for plotting. Either "static" or "interactive" (Default = "static").
- `label.sep` Separator used for label merging (Default = ", ").
geneViewGuide

width

Width of the plot in cm. Defaults to minimal size for readable labels and supports reactive.

height

Height of the plot in cm. Defaults to minimal size for readable labels and supports reactive.

ppi

Pixel per inch. Defaults to 72 and supports reactive.

scale

Scale plot size. Defaults to 1, supports reactive.

Details

Width/ height/ ppi less or equal to default will use default value.

Ppi less or equal to zero will use default.

Value

Reactive containing data.table used for plotting.

geneViewGuide geneView module guide

Description

geneView module guide

Usage

geneViewGuide(session)

Arguments

session The shiny session

Value

A shiny reactive that contains the texts for the Guide steps.
geneViewUI

geneView’s module UI representation

Description

geneView’s module UI representation

Usage

geneViewUI(id, plot.columns = 3)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>The ID of the modules namespace.</td>
</tr>
<tr>
<td>plot.columns</td>
<td>Initial value of plot column slider. Integer value between 1 and 7 (Default = 3).</td>
</tr>
</tbody>
</table>

Value

A list with HTML tags from tag.

global_cor_heatmap

global correlation heatmap module server logic

Description

global correlation heatmap module server logic

Usage

global_cor_heatmap(input, output, session, clarion,
                   plot.method = "static", width = "auto", height = "auto",
                   ppi = 72, scale = 1)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>input</td>
<td>Shiny’s input object</td>
</tr>
<tr>
<td>output</td>
<td>Shiny’s output object</td>
</tr>
<tr>
<td>session</td>
<td>Shiny’s session object</td>
</tr>
<tr>
<td>clarion</td>
<td>A clarion object. See Clarion. (Supports reactive)</td>
</tr>
<tr>
<td>plot.method</td>
<td>Choose which method is used for plotting. Either &quot;static&quot; or &quot;interactive&quot; (Default = &quot;static&quot;).</td>
</tr>
<tr>
<td>width</td>
<td>Width of the plot in cm. Defaults to minimal size for readable labels and supports reactive.</td>
</tr>
</tbody>
</table>
Value

Reactive containing data used for plotting.

---

global_cor_heatmapUI  global correlation heatmap module UI representation

Description

global correlation heatmap module UI representation

Usage

global_cor_heatmapUI(id)

Arguments

id  The ID of the modules namespace.

Value

A list with HTML tags from tag

---

global_cor_heatmap_guide  global correlation heatmap module guide

Description

global correlation heatmap module guide

Usage

global_cor_heatmap_guide(session)

Arguments

session  The shiny session

Value

A shiny reactive that contains the texts for the Guide steps.
heatmap

heatmap

Description
heatmap module server logic

Usage
heatmap(input, output, session, clarion, plot.method = "static",
label.sep = ", ", width = "auto", height = "auto", ppi = 72,
scale = 1)

Arguments
input Shiny’s input object
output Shiny’s output object
session Shiny’s session object
clarion A clarion object. See Clarion. (Supports reactive)
plot.method Choose which method is used for plotting. Either "static" or "interactive" (Default = "static").
label.sep Separator used for label merging (Default = ", ").
width Width of the plot in cm. Defaults to minimal size for readable labels and supports reactive.
height Height of the plot in cm. Defaults to minimal size for readable labels and supports reactive.
ppi Pixel per inch. Defaults to 72 and supports reactive.
scale Scale plot size. Defaults to 1, supports reactive.

Value
Reactive containing data used for plotting.

heatmapGuide

heatmapGuide

Description
heatmap module guide

Usage
heatmapGuide(session)
Arguments

session The shiny session

Value

A shiny reactive that contains the texts for the Guide steps.

heatmapUI heatmap module UI representation

Description

heatmap module UI representation

Usage

heatmapUI(id, row.label = TRUE)

Arguments

id The ID of the modules namespace.
row.label Boolean Value set initial Value for rowlabel checkbox (Default = TRUE).

Value

A list with HTML tags from tag.

label label module server logic

Description

label module server logic

Usage

label(input, output, session, data, label = "Select label columns", multiple = TRUE, sep = ",", unique = TRUE, unique_sep = ",", disable = NULL)
**labelUI**

**Arguments**

- **input**  
  Shiny’s input object.

- **output**  
  Shiny’s output object.

- **session**  
  Shiny’s session object.

- **data**  
  Data.table used for label creation. Column names will be used for selection. (supports reactive)

- **label**  
  Set label of selectizeInput.

- **multiple**  
  Allow multiple selection which will be merged with sep (default = TRUE).

- **sep**  
  Separator used to collapse selection (default = ", ").

- **unique**  
  Make labels unique. Defaults to TRUE. See make.unique.

- **unique_sep**  
  Separator used for unique (default = "_"). Should differ from sep.

- **disable**  
  Reactive containing boolean. To disable/enable module.

**Value**

Reactive containing list(label = vector of strings or NULL on empty selection, selected = user input).

---

**labelUI** label module UI representation

**Description**

label module UI representation

**Usage**

labelUI(id)

**Arguments**

- **id**  
  The ID of the modules namespace

**Value**

A list with HTML tags from tag
limit

limit module server logic

Description
limit module server logic

Usage
limit(input, output, session, lower = NULL, upper = NULL)

Arguments
input  Shiny’s input object.
output Shiny’s output object.
session Shiny’s session object.
lower  Set lower limit (supports reactive).
upper  Set upper limit (supports reactive).

Value
Reactive containing: list(lower, upper).

limitUI

limit module UI representation

Description
limit module UI representation

Usage
limitUI(id, label = "Limit")

Arguments
id The ID of the modules namespace
label Set the modules label.

Value
A list with HTML tags from tag
log_message

Description

logger message convenience function

Usage

log_message(message, level = c("DEBUG", "INFO", "WARN", "ERROR", "FATAL"), token = NULL)

Arguments

message String of message to be written in log. See levellog.
level Set priority level of the message (number or character). See levellog.
token Use token bound to this identifier.

Details

Does nothing if logger doesn’t exist.

marker

Description

marker module server logic

Usage

marker(input, output, session, clarion)

Arguments

input Shiny’s input object.
output Shiny’s output object.
session Shiny’s session object.
clarion A clarion object. See Clarion. (Supports reactive)

Value

A named list containing reactives (highlight, color, labelColumn, label, clarion).
markerUI

**marker module UI representation**

### Description

marker module UI representation

### Usage

```r
description = label = "Highlight/Label Selected Features"
```

### Arguments

- **id**
  - The ID of the modules namespace
- **label**
  - Set label of first element.

### Value

- A list with HTML tags from `tag`

### orNumeric

**orNumeric module server logic**

### Description

Provides server logic for the orNumeric module.

### Usage

```r
orNumeric(input, output, session, choices, value, label = "Column",
          step = 100, stepsize = NULL,
          min. = shiny::reactive(min(choices_r(), na.rm = TRUE)),
          max. = shiny::reactive(max(choices_r(), na.rm = TRUE)),
          label.slider = NULL, zoomable = TRUE, reset = NULL)
```

### Arguments

- **input**
  - Shiny’s input object.
- **output**
  - Shiny’s output object.
- **session**
  - Shiny’s session object.
- **choices**
  - A list or a numeric vector with the possible choices offered in the UI. See `sliderInput` (Supports reactive).
- **value**
  - Initial value of the slider. Creates a ranged slider if numeric vector of two given (Supports reactive).
orNumericUI

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>label</td>
<td>Label of the entire module.</td>
</tr>
<tr>
<td>step</td>
<td>Number of steps on interval (Default = 100).</td>
</tr>
<tr>
<td>stepsize</td>
<td>Value defining interval size of the slider. Will be used instead of step (Default = NULL).</td>
</tr>
<tr>
<td>min.</td>
<td>Minimum value that can be selected on slider (defaults to min(choices)) (Supports reactive).</td>
</tr>
<tr>
<td>max.</td>
<td>Maximum value that can be selected on slider (defaults to max(choices)) (Supports reactive).</td>
</tr>
<tr>
<td>label.slider</td>
<td>A character vector of length one with the label for the sliderInput.</td>
</tr>
<tr>
<td>zoomable</td>
<td>Boolean to enable zooming. Redefine the sliders range. Defaults to TRUE.</td>
</tr>
<tr>
<td>reset</td>
<td>A reactive which will trigger a module reset on change.</td>
</tr>
</tbody>
</table>

Value

Returns a reactive containing a named list with the label, the selected choices as a character vector (text), a boolean vector of length length(choices) (bool), and a vector of the selected value(s) (value), indicating whether a item has been chosen. If no item has been chosen, the return is TRUE for items.

---

orNumericUI | orNumeric module UI representation

Description

This module allows to select value/range inputs from a sliderInput element. The functions creates HTML tag definitions of its representation based on the parameters supplied.

Usage

orNumericUI(id)

Arguments

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>The ID of the modules namespace.</td>
</tr>
</tbody>
</table>

Value

A list with HTML tags from tag.
**orTextual**  

**orTextual module server logic**

**Description**

Provides server logic for the orTextual module.

**Usage**

```
orTextual(input, output, session, choices, selected = NULL,  
label = "Column", delimiter = NULL, multiple = TRUE,  
contains = FALSE, reset = NULL)
```

**Arguments**

- **input**  
  Shiny’s input object.
- **output**  
  Shiny’s output object.
- **session**  
  Shiny’s session object.
- **choices**  
  A list or a character vector with the possible choices offered in the UI. See `selectInput`.
- **selected**  
  The initially selected value. See `selectInput`.
- **label**  
  A character vector of length one with the label for the `selectInput`.
- **delimiter**  
  A single character indicating if and how items are delimited (default: NULL indicates no delimitation). Only if contains = FALSE.
- **multiple**  
  Whether or not selection of multiple items is allowed.
- **contains**  
  Logical variable. If TRUE shows module as a textsearch input.
- **reset**  
  A reactive which will trigger a module reset on change.

**Value**

Returns a reactive containing a named list with the label, the selected choices as a character vector (text) and a boolean vector of length length(Choices) (bool), indicating whether a item has been chosen. If no item has been chosen, the return is TRUE for items.
orTextualUI

orTextualUI  orTextual module UI representation

Description
This module allows to select (multiple) inputs from a `selectInput` element. The functions creates HTML tag definitions of its representation based on the parameters supplied.

Usage
```r
orTextualUI(id)
```

Arguments
- `id` The ID of the modules namespace.

Value
A list with HTML tags from `tag`.

parser

Method to parse input file.

Description
Method to parse input file.

Usage
```r
parser(file, dec = ".")
```

Arguments
- `file` Path to file that needs parsing.
- `dec` The decimal separator. See `fread`.

Value
Clarion object. See `Clarion`
parse_MaxQuant

Converting MaxQuant Output file proteinGroups.txt to CLARION format by creating a headline of metadata for each column

Description

List with columns of reduced version (see config.json file) If you only want the samples of a specific keyword write: column;exp For example: You got: Intensity Intensity 'experiment_name' Do you want both add "Intensity" to the list. Do you only want the sample add "Intensity;exp" to the list Anything else like 'Intensity;ex' or 'Intensity;' results in writing both. Only works if there are samples of that type. If not, column does not show up in file

Usage

parse_MaxQuant(proteinGroups_in, summary_in, outfile, outfile_reduced, 
config = system.file("extdata", "parser_MaxQuant_config.json", package 
= "wilson"), delimiter = ";", format = NULL, version = NULL, 
experiment_id = NULL)

Arguments

proteinGroups_in       path of proteinGroup.txt file
summary_in             path of belonging summary.txt file
outfile                path of full CLARION output file
outfile_reduced       path of reduced CLARION output file
config                 path of config file (containing information about metadata)
delimiter              delimiter (Default = ;)
format                 pre-header information about format (optional)
version                pre-header information about version (optional)
experiment_id         pre-header information about experiment id (optional)

Value

TRUE on success

Author(s)

Rene Wiegandt
Description

pca module server logic

Usage

pca(input, output, session, clarion, width = 28, height = 28, ppi = 72, scale = 1)

Arguments

input  Shiny’s input object
output Shiny’s output object
session Shiny’s session object
clarion A clarion object. See Clarion. (Supports reactive)
width  Width of the plot in cm. Defaults to 28 and supports reactive.
height Height of the plot in cm. Defaults to 28 and supports reactive.
ppi  Pixel per inch. Defaults to 72 and supports reactive.
scale Scale plot size. Defaults to 1, supports reactive.

Details

Width/ height/ ppi less or equal to zero will use default value.

Value

A reactive containing list with dimensions.

Description

pca module guide

Usage

pcaGuide(session)
Arguments

session The shiny session

Value

A shiny reactive that contains the texts for the Guide steps.

---

c pcaUI  

*pca module UI representation*

Description

pca module UI representation

Usage

pcaUI(id, show.label = TRUE)

Arguments

id The ID of the modules namespace.
show.label Set initial value of show label checkbox (Default = TRUE).

Value

A list with HTML tags from tag.

---

release_questions

*Defines additional questions asked before CRAN submission. DO NOT EXPORT!*

Description

Defines additional questions asked before CRAN submission. DO NOT EXPORT!

Usage

release_questions()
scatterPlot

scatterPlot module server logic

Description

scatterPlot module server logic

Usage

scatterPlot(input, output, session, clarion, marker.output = NULL, plot.method = "static", width = "auto", height = "auto", ppi = 72, scale = 1)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>input</td>
<td>Shiny’s input object</td>
</tr>
<tr>
<td>output</td>
<td>Shiny’s output object</td>
</tr>
<tr>
<td>session</td>
<td>Shiny's session object</td>
</tr>
<tr>
<td>clarion</td>
<td>A clarion object. See Clarion. (Supports reactive)</td>
</tr>
<tr>
<td>marker.output</td>
<td>Marker module output. See marker.</td>
</tr>
<tr>
<td>plot.method</td>
<td>Choose to rather render a ’interactive’ or ’static’ plot. Defaults to ’static’</td>
</tr>
<tr>
<td>width</td>
<td>Width of the plot in cm. Defaults to minimal size for readable labels and supports reactive.</td>
</tr>
<tr>
<td>height</td>
<td>Height of the plot in cm. Defaults to minimal size for readable labels and supports reactive.</td>
</tr>
<tr>
<td>ppi</td>
<td>Pixel per inch. Defaults to 72 and supports reactive.</td>
</tr>
<tr>
<td>scale</td>
<td>Scale plot size. Defaults to 1, supports reactive.</td>
</tr>
</tbody>
</table>

Details

As markerOutput provides a second dataset used for highlighting it is crucial for it to have the same columnnames as the dataset provided by clarion.

Intersections between marker and clarion will be removed from clarion in favor of highlighting them.

Value

Returns reactive containing data used for plot.
scatterPlotGuide  

**Description**
scatterPlot module guide

**Usage**
scatterPlotGuide(session, marker = FALSE)

**Arguments**
- **session**: The shiny session
- **marker**: Logical if marker step should be enabled (Default = FALSE).

**Value**
A shiny reactive that contains the texts for the Guide steps.

scatterPlotUI  

**Description**
scatterPlot module UI representation

**Usage**
scatterPlotUI(id)

**Arguments**
- **id**: The ID of the modules namespace.

**Value**
A list with HTML tags from `tag`.
searchData  

Function to search data for selection

Description

Function to search data for selection

Usage

searchData(input, choices, options = c("=" , "<" , ", >"),
  min. = min(choices, na.rm = TRUE), max. = max(choices, na.rm = TRUE))

Arguments

input Vector length one (single) or two (ranged) containing numeric values for selection.
choices Vector on which input values are applied.
options Vector on how the input and choices should be compared. It can contain: single = c(="", "<", ", >") or ranged = c("inner", "outer").
min. Minimum value that can be selected on slider (defaults to min(choices)).
max. Maximum value that can be selected on slider (defaults to max(choices)).

Value

Returns a logical vector with the length of choices, where every matched position is TRUE.

sequentialPalettes  

Function to generate sequential (one-sided) color palettes (e.g. for expression, enrichment)

Description

Function to generate sequential (one-sided) color palettes (e.g. for expression, enrichment)

Usage

sequentialPalettes(n)

Arguments

n Number of colors to generate

Value

A data.table with (named) color palettes of length n
set_logger  
set a log4r logger used within the package

Description
set a log4r logger used within the package

Usage
set_logger(logger, token = NULL)

Arguments
logger  
A logger object see create.logger. NULL to disable logging.
token  
Set a unique identifier for this logger.

Details
This function will save each logger in the wilson.globals environment. Each logger is stored by the
name 'logger'[token] (e.g. 'logger6b821824b0b53b1a3e8f531a34d0d6e6').
Use onSessionEnded to clean up after logging. See onFlush.

tobias_parser  
TOBIAS TFBS table to clarion parser

Description
Click here for more information about TOBIAS.

Usage
tobias_parser(input, output, filter_columns = NULL,
filter_pattern = NULL, config = system.file("extdata",
"tobias_config.json", package = "wilson"), omit_NA = FALSE,
condition_names = NULL, condition_pattern = ".bound$",
in_field_delimiter = ",", dec = ".", ...)

Arguments
input  
Path to input table
output  
Output path.
filter_columns  
Either a vector of columnnames or a file containing one columnname per row.
filter_pattern  
Keep columns matching the given pattern. Uses parameter filter_columns for
matching if set. In the case of no matches a warning will be issued and all
columns will be used.
transformation

config Json file containing metadata information for all columns. Will use first occurrence for duplicate column names.

omit NA Logical whether all rows containing NA should be removed.

condition_names Vector of condition names. Default = NULL. Used to classify columns not provided in config.

condition_pattern Used to identify condition names by matching and removing given pattern with grep. Ignored when condition_names is set.

in_field_delimiter Delimiter for multi value fields. Default = ‘,’.

dec Decimal separator. Used in file reading and writing.

... Used as header information.

Details

During conversion the parser will try to use the given config (if provided) to create the Clarion metadata. In the case of insufficient config information it will try to approximate by referencing condition names issuing warnings in the process.

As the format requires an unique id the parser will create one if necessary.

Factor grouping (metadata factor columns) is currently not implemented!

transformation

transformation module server logic

Description

The module provides several transformations on a numeric data matrix for the user.

Usage

transformation(input, output, session, data, transpose = FALSE, pseudocount = 1, replaceInf = TRUE, replaceNA = TRUE)

Arguments

input Shiny’s input object.
output Shiny’s output object.
session Shiny’s session object.
data Numeric matrix on which transformation is performed (column-wise). (Supports reactive)
transpose Whether the matrix should be transposed to enable row-wise transformation. (Supports reactive)
transformationUI

pseudocount  Numeric Variable to add a pseudocount to log-based transformations. (Supports reactive)
replaceInf   Change Infinite to NA, applied after transformation. (Supports reactive)
replaceNA    Change NA to 0, applied after transformation. (Supports reactive)

Value
Named list of two containing data and name of the used method. data: Reactive containing the transformed matrix. Infinite values are replaced by NA and NA values are replaced by 0. method: Reactive containing String. transpose: Reactive containing String.

description
This function provides an input to select a transformation method.

Usage
transformationUI(id, label = "Transformation", selected = "raw", choices = list(None = "raw", log2 = "log2", \'-log2\' = "-log2", log10 = "log10", \'-log10\' = "-log10", \'Z score\' = "zscore", \'regularized log\' = "rlog"), transposeOptions = FALSE)

Arguments
id               The ID of the modules namespace.
label            A character vector of length one with the label for the selectInput.
selected         The initially selected value. See selectInput.
choices          Named list of available transformations. Possible transformations are list('None' = "raw", 'log2' = "log2", "-log2" = "-log2", 'log10' = "log10", "-log10" = "-log10", 'Z score' = "zscore", 'regularized log' = "rlog") which is also the default.
transposeOptions Boolean value if transpose radioButtons are shown (Default = FALSE).

Value
A list with HTML tags from tag.
Index

**Topic datasets**
- Clarion, 5
- and, 3
- andUI, 4
- categoricalPalettes, 4
- Clarion, 5, 15, 17, 19, 21, 25, 29, 31, 33
- colorPicker, 6
- colorPickerUI, 7
- columnSelector, 7
- columnSelectorUI, 8
- create, logger, 36
- create_geneview, 9
- create_heatmap, 10
- create_pca, 11
- create_scatterplot, 12
- divergingPalettes, 13
- download, 14
- downloadHandler, 14
- equalize, 14
- featureSelector, 15
- featureSelectorGuide, 16
- featureSelectorUI, 16
- force, 17
- forceAndCall, 17
- forceArgs, 17
- fread, 29
- geneView, 17
- geneViewGuide, 18
- geneViewUI, 19
- geom_point, 12
- geom_text, 12
- get, 17
- global_cor_heatmap, 19
- global_cor_heatmap_guide, 20
- global_cor_heatmapUI, 20
- grep, 37
- heatmap, 21
- heatmapGuide, 21
- heatmapr, 10
- heatmapUI, 22
- label, 22
- labelUI, 23
- levellog, 25
- limit, 24
- limitUI, 24
- lims, 13
- log_message, 25
- make_unique, 23
- marker, 25, 33
- markerUI, 26
- match.call, 17
- onFlush, 36
- orNumeric, 26
- orNumericUI, 27
- orTextual, 28
- orTextualUI, 29
- parse_MaxQuant, 30
- parser, 29
- pca, 31
- pcaGuide, 31
- pcaUI, 32
- points, 11
- release_questions, 32
- scatterPlot, 33
- scatterPlotGuide, 34
- scatterPlotUI, 34
- searchData, 35
- selectInput, 28, 29, 38
- sequentialPalettes, 35
set_logger, 36
sliderInput, 26, 27

tag, 4, 7, 8, 16, 19, 20, 22–24, 26, 27, 29, 32, 34, 38
tobias_parser, 36
toJSON, 14
transformation, 37
transformationUI, 38

zipr, 14