

# Package ‘wilson’

May 26, 2020

**Type** Package

**Title** Web-Based Interactive Omics Visualization

**Version** 2.4.0

**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, zip, shinyWidgets

**RoxygenNote** 7.1.0

**biocViews**

**Suggests** knitr, rmarkdown, testthat, stringi, utils

**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** 2020-05-26 11:10:02 UTC

**R topics documented:**

and	3
andUI	4
categoricalPalettes	5
Clarion	5
colorPicker	8
colorPickerUI	9
columnSelector	10
columnSelectorUI	11
create_geneview	12
create_heatmap	13
create_pca	14
create_scatterplot	16
divergingPalettes	17
download	18
equalize	19
featureSelector	19
featureSelectorGuide	20
featureSelectorUI	21
forceArgs	21
geneView	22
geneViewGuide	23
geneViewUI	23
global_cor_heatmap	24
global_cor_heatmapUI	25
global_cor_heatmap_guide	25
heatmap	26
heatmapGuide	27
heatmapUI	27
install_app	28
label	28
labelUI	29
limit	30
limitUI	30
log_message	31
marker	31
markerUI	32
orNumeric	32
orNumericUI	33
orTextual	34
orTextualUI	35
parser	35
parse_MaxQuant	36
pca	37
pcaGuide	38
pcaUI	38
release_questions	39

scatterPlot . . . . .	39
scatterPlotGuide . . . . .	40
scatterPlotUI . . . . .	41
searchData . . . . .	41
sequentialPalettes . . . . .	42
set_logger . . . . .	42
tobias_parser . . . . .	43
transformation . . . . .	44
transformationUI . . . . .	45

<b>Index</b>	<b>46</b>
--------------	-----------

---

and	<i>AND module server logic</i>
-----	--------------------------------

---

## Description

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

## Usage

```
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)

<code>element.grouping</code>	Group features in boxes. (Data.table: first column = columnnames, second column = groupnames) (Supports reactive)
<code>column.labels</code>	Additional labels for the columns, defaults to names(data).
<code>delimiter</code>	A single character, or a vector indicating how column values are delimited. (Fills vector sequentially if needed)(Supports reactive)
<code>multiple</code>	Whether or not textual ORs should allow multiple selections. (Fills vector sequentially if needed)(Supports reactive)
<code>contains</code>	Whether or not textual ORs are initialized as textInput checking entries for given string. (Fills vector sequentially if needed)(Supports reactive)
<code>ranged</code>	Whether or not numeric ORs are ranged. (Fills vector sequentially if needed)(Supports reactive)
<code>step</code>	Set numeric ORs slider steps. (Fills vector sequentially if needed)(Supports reactive)
<code>reset</code>	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                      *Clarion R6-class definition*

---

**Description**

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

**Constructor**

```
Clarion$new(header = NULL, metadata, data, validate = TRUE)
```

**Constructor Arguments**

<b>Variable</b>	<b>Return</b>
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.

**Public 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

### Public methods:

- `Clarion$get_id()`
- `Clarion$get_name()`
- `Clarion$get_delimiter()`
- `Clarion$is_delimited()`
- `Clarion$get_factors()`
- `Clarion$get_level()`
- `Clarion$get_label()`
- `Clarion$validate()`
- `Clarion$new()`
- `Clarion$write()`
- `Clarion$clone()`

**Method** `get_id()`: Returns name of unique identifier column. Assumes first feature to be unique if not specified.

*Usage:*

```
Clarion$get_id()
```

*Returns:* Name of the id column.

**Method** `get_name()`: Returns name of name column. If not specified return unique Id.

*Usage:*

```
Clarion$get_name()
```

*Returns:* Name of the name column.

**Method** `get_delimiter()`: Return delimiter used within multi-value fields (no delimiter = NULL).

*Usage:*

```
Clarion$get_delimiter()
```

**Method** `is_delimited()`: Logical whether the given column name is delimited.

*Usage:*

```
Clarion$is_delimited(x)
```

*Arguments:*

x Name of the column.

*Returns:* boolean

**Method** `get_factors()`: Get factors to all columns.

*Usage:*

```
Clarion$get_factors()
```

*Details:* Named factors (e.g. factor1="name") will be cropped to their name.

*Returns:* Returns a data.table columns: key and factor(s) if any.

**Method** `get_level()`: Get level(s) to given column name(s).

*Usage:*

```
Clarion$get_level(column)
```

*Arguments:*

column One or more column name(s).

*Returns:* Provide a vector of levels to the given columnnames in column. Returns NA for missing columns and character(0) if column = NULL.

**Method** `get_label()`: Get label(s) to given column name(s).

*Usage:*

```
Clarion$get_label(column = NULL, sub_label = TRUE, sep = " ")
```

*Arguments:*

column One or more column name(s).

sub\_label Whether the sub\_label should be included.

sep Separator between label and sub\_label.

*Details:* If a column does not have a label the key is returned.

*Returns:* Provides a vector of labels (+ sub\_label) to the given columnnames in column. Returns NA for missing columns and all labels if column = NULL.

**Method** `validate()`: Check the object for inconsistencies.

*Usage:*

```
Clarion$validate(solve = TRUE)
```

*Arguments:*

solve For solve = TRUE try to resolve some warnings.

**Method** `new()`: Initialize a new clarion object.

*Usage:*

```
Clarion$new(header = NULL, metadata, data, validate = TRUE)
```

*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.

*Returns:* Clarion object.

**Method** `write()`: Save the object as a clarion file.

*Usage:*

```
Clarion$write(file)
```

*Arguments:*

file Filename for the file to be written.

**Method** clone(): The objects of this class are cloneable with this method.

*Usage:*

```
Clarion$clone(deep = FALSE)
```

*Arguments:*

deep Whether to make a deep clone.

### Examples

```
## 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)
```

---

colorPicker

*colorPicker module server logic*

---

### Description

Provides server logic for the colorPicker2 module.

### Usage

```
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	<i>colorPicker module UI representation</i>
---------------	---

---

### Description

The functions 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

```
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 drop-down, 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 colorpalettes 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

```
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	<i>columnSelector module UI representation</i>
------------------	--

---

**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.

---

create\_geneview      *Method for geneView creation*

---

## Description

Method for geneView creation

## Usage

```
create_geneview(  
  data,  
  grouping,  
  plot.type = "line",  
  facet.target = "gene",  
  facet.cols = 2,  
  colors = NULL,  
  ylabel = NULL,  
  ylimits = 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	<i>Method for heatmap creation</i>
----------------	------------------------------------

---

**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. <code>c("none", "both", "column", "row")</code>
clustdist	Which cluster distance to use. See <a href="#">heatmapr</a> .
clustmethod	Which cluster method to use. See <a href="#">heatmapr</a> .
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`.

### 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`.

---

<code>create_pca</code>	<i>Method for pca creation.</i>
-------------------------	---------------------------------

---

### 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

<code>data</code>	data.table from which the plot is created (First column will be handled as row-names if not numeric).
<code>color.group</code>	Vector of groups according to samples (= column names).
<code>color.title</code>	Title of the color legend.
<code>palette</code>	Vector of colors used for color palette.
<code>shape.group</code>	Vector of groups according to samples (= column names).
<code>shape.title</code>	Title of the shape legend.
<code>shapes</code>	Vector of shapes see <a href="#">points</a> . Will recycle/ cut off shapes if needed. Default = <code>c(15:25)</code>
<code>dimension.a</code>	Number of dimension displayed on X-Axis.
<code>dimension.b</code>	Number of dimension displayed on Y-Axis.
<code>dimensions</code>	Number of dimensions to create.
<code>on.columns</code>	Boolean perform pca on columns or rows.
<code>labels</code>	Boolean show labels.
<code>custom.labels</code>	Vector of custom labels. Will replace columnnames.
<code>pointsize</code>	Size of the data points.
<code>labelsize</code>	Size of texts inside plot (default = 3).
<code>width</code>	Set the width of the plot in cm (default = 28).
<code>height</code>	Set the height of the plot in cm (default = 28).
<code>ppi</code>	Pixel per inch (default = 72).
<code>scale</code>	Modify plot size while preserving aspect ratio (Default = 1).

### 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).

---

create\_scatterplot      *Method for scatter plot creation*

---

## Description

Method for scatter plot creation

## Usage

```
create_scatterplot(  
  data,  
  data.labels = NULL,  
  data.hovertext = NULL,  
  transparency = 1,  
  pointsize = 1,  
  labelsize = 3,  
  color = NULL,  
  x_label = "",  
  y_label = "",  
  z_label = "",  
  density = TRUE,  
  line = TRUE,  
  categorized = FALSE,  
  highlight.data = NULL,  
  highlight.labels = NULL,  
  highlight.hovertext = NULL,  
  highlight.color = "#FF0000",  
  xlim = NULL,  
  ylim = NULL,  
  colorbar.limits = NULL,  
  width = "auto",  
  height = "auto",  
  ppi = 72,  
  plot.method = "static",  
  scale = 1  
)
```

## Arguments

data	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 <a href="#">geom_point</a> .
pointsize	Set point size. See <a href="#">geom_point</a> .
labelsize	Set label size. See <a href="#">geom_text</a> .



<code>color</code>	Vector of colors used for color palette.
<code>x_label</code>	Label x-Axis
<code>y_label</code>	Label Y-Axis
<code>z_label</code>	Label Z-Axis
<code>density</code>	Boolean value, perform 2d density estimate.
<code>line</code>	Boolean value, add reference line.
<code>categorized</code>	Z-Axis (if exists) as categories.
<code>highlight.data</code>	data.table containing data to highlight. Same structure as data.
<code>highlight.labels</code>	Vector of labels used for highlighted data. Length has to be equal to <code>nrow(highlight.data)</code> .
<code>highlight.hovertext</code>	Character vector with additional hovertext. Length has to be equal to <code>nrow(highlight.data)</code> .
<code>highlight.color</code>	String with hexadecimal color-code.
<code>xlim</code>	Numeric vector of two setting min and max limit of x-axis. See <a href="#">lims</a> .
<code>ylim</code>	Numeric vector of two setting min and max limit of y-axis. See <a href="#">lims</a> .
<code>colorbar.limits</code>	Vector with min, max values for colorbar (Default = NULL).
<code>width</code>	Set plot width in cm (Default = "auto").
<code>height</code>	Set plot height in cm (Default = "auto").
<code>ppi</code>	Pixel per inch (default = 72).
<code>plot.method</code>	Whether the plot should be 'interactive' or 'static' (Default = 'static').
<code>scale</code>	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	<i>Function used for downloading. Creates a zip container containing plot in png, pdf and user input in json format. Use inside <a href="#">downloadHandler</a> content function.</i>
----------	---

---

**Description**

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**

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

**Arguments**

file	See <a href="#">downloadHandler</a> content parameter.
filename	See <a href="#">downloadHandler</a> .
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 <a href="#">toJSON</a>

**Value**

Path to zip archive invisibly. See [zipr](#).

---

equalize	<i>Method to get equalized min/max values from vector</i>
----------	---

---

**Description**

Method to get equalized min/max values from vector

**Usage**

```
equalize(values)
```

**Arguments**

values	Numeric vector or table
--------	-------------------------

**Value**

Vector with c(min, max).

---

featureSelector	<i>featureSelector module server logic</i>
-----------------	--

---

**Description**

featureSelector module server logic

**Usage**

```
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 <a href="#">Clarion</a> . (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	<i>featureSelector module UI representation</i>
-------------------	---

---

**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	<i>Force evaluation of the parent function's arguments.</i>
-----------	---

---

**Description**

Force evaluation of the parent function's arguments.

**Usage**

```
forceArgs(args)
```

**Arguments**

args	List of Argument names to force evaluation. Defaults to all named arguments see <a href="#">match.call</a> .
------	--

**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

```
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 <a href="#">Clarion</a> . (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.

## 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	<i>geneView module guide</i>
---------------	------------------------------

---

**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	<i>geneView's module UI representation</i>
------------	--

---

**Description**

geneView's module UI representation

**Usage**

```
geneViewUI(id, plot.columns = 3)
```

**Arguments**

id	The ID of the modules namespace.
plot.columns	Initial value of plot column slider. Integer value between 1 and 7 (Default = 3).

**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

input	Shiny's input object
output	Shiny's output object
session	Shiny's session object
clarion	A clarion object. See <a href="#">Clarion</a> . (Supports reactive)
plot.method	Choose which method is used for plotting. Either "static" or "interactive" (Default = "static").
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.



---

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 module server logic*

---

**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**

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

**Value**

Reactive containing data used for plotting.

---

heatmapGuide	<i>heatmap module guide</i>
--------------	-----------------------------

---

**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	<i>heatmap module UI representation</i>
-----------	---

---

**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](#).

---

install_app	<i>Download and install Wilson Apps</i>
-------------	---

---

**Description**

Download and install Wilson Apps

**Usage**

```
install_app(  
  location = ".",  
  remove_data = FALSE,  
  start_after_install = FALSE,  
  app_name = "wilson-basic",  
  repository = "https://github.molgen.mpg.de/loosolab/wilson-apps"  
)
```

**Arguments**

location	Where the app should be installed. Default is current location.
remove_data	If TRUE demo data will be deleted.
start_after_install	Start the app when done installing.
app_name	Select app to install.
repository	Link to the repository that holds the apps.

**Details**

Will create a folder named after parameter app\_name.

---

label	<i>label module server logic</i>
-------	----------------------------------

---

**Description**

label module server logic

**Usage**

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

**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 <a href="#">make.unique</a> .
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	<i>label module UI representation</i>
---------	---------------------------------------

---

**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	<i>limit module server logic</i>
-------	----------------------------------

---

**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	<i>limit module UI representation</i>
---------	---------------------------------------

---

**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	<i>logger message convenience function</i>
-------------	--

---

**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 <a href="#">levellog</a> .
level	Set priority level of the message (number or character). See <a href="#">levellog</a> .
token	Use token bound to this identifier.

**Details**

Does nothing if logger doesn't exist.

---

marker	<i>marker module server logic</i>
--------	-----------------------------------

---

**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 <a href="#">Clarion</a> . (Supports reactive)

**Value**

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

---

markerUI	<i>marker module UI representation</i>
----------	--

---

**Description**

marker module UI representation

**Usage**

```
markerUI(id, 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	<i>orNumeric module server logic</i>
-----------	--------------------------------------

---

**Description**

Provides server logic for the orNumeric module.

**Usage**

```
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 <a href="#">sliderInput</a> (Supports reactive).
value	Initial value of the slider. Creates a ranged slider if numeric vector of two given (Supports reactive).
label	Label of the entire module.
step	Number of steps on interval (Default = 100).
stepsize	Value defining interval size of the slider. Will be used instead of step (Default = NULL).
min.	Minimum value that can be selected on slider (defaults to min(choices)) (Supports reactive).
max.	Maximum value that can be selected on slider (defaults to max(choices)) (Supports reactive).
label.slider	A character vector of length one with the label for the <a href="#">sliderInput</a> .
zoomable	Boolean to enable zooming. Redefine the sliders range. Defaults to TRUE.
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), 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**

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

**Value**

A list with HTML tags from [tag](#).

---

orTextual	<i>orTextual module server logic</i>
-----------	--------------------------------------

---

### 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,
  parse_mode = TRUE
)
```

### 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 <a href="#">selectInput</a> .
selected	The initially selected value. See <a href="#">selectInput</a> .
label	A character vector of length one with the label for the <a href="#">selectInput</a> .
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.
parse_mode	Boolean to enable text to selection parsing. Ignored if multiple = FALSE or contains = TRUE.

### 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	<i>orTextual module UI representation</i>
-------------	---

---

**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**

```
orTextualUI(id)
```

**Arguments**

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

**Value**

A list with HTML tags from [tag](#).

---

parser	<i>Method to parse input file.</i>
--------	------------------------------------

---

**Description**

Method to parse input file.

**Usage**

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

**Arguments**

file	Path to file that needs parsing.
dec	The decimal separator. See <a href="#">fread</a> .

**Value**

Clarion object. See [Clarion](#)

---

parse_MaxQuant	<i>Converting MaxQuant Output file proteinGroups.txt to CLARION format by creating a headline of metadata for each column</i>
----------------	---

---

### 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

---

pca

*pca module server logic*

---

**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 <a href="#">Clarion</a> . (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.

---

pcaGuide	<i>pca module guide</i>
----------	-------------------------

---

**Description**

pca module guide

**Usage**

```
pcaGuide(session)
```

**Arguments**

session	The shiny session
---------	-------------------

**Value**

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

---

pcaUI	<i>pca module UI representation</i>
-------	-------------------------------------

---

**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	<i>Defines additional questions asked before CRAN submission. DO NOT EXPORT!</i>
-------------------	--

---

**Description**

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

**Usage**

```
release_questions()
```

---

scatterPlot	<i>scatterPlot module server logic</i>
-------------	--

---

**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**

input	Shiny's input object
output	Shiny's output object
session	Shiny's session object
clarion	A clarion object. See <a href="#">Clarion</a> . (Supports reactive)
marker.output	Marker module output. See <a href="#">marker</a> .
plot.method	Choose to rather render a 'interactive' or 'static' plot. Defaults to 'static'.
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

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      *scatterPlot module guide*

---

### 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	<i>scatterPlot module UI representation</i>
---------------	---

---

**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	<i>Function to search data for selection</i>
------------	--

---

**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	<i>Function to generate sequential (one-sided) color palettes (e.g. for expression, enrichment)</i>
--------------------	---

---

**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	<i>set a log4r logger used within the package</i>
------------	---

---

**Description**

set a log4r logger used within the package

**Usage**

```
set_logger(logger, token = NULL)
```

**Arguments**

logger	A logger object see <a href="#">create.logger</a> . 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	<i>TOBIAS TFBS table to clarion parser</i>
---------------	--

---

## 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.
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 <a href="#">grep</a> . 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	<i>transformation module server logic</i>
----------------	---

---

**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)
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**

Namedlist 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.

---

transformationUI	<i>transformation module UI representation</i>
------------------	--

---

### 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 <a href="#">selectInput</a> .
selected	The initially selected value. See <a href="#">selectInput</a> .
choices	Named list of available transformations. Possible transformations are list( <code>'None'</code> = "raw", <code>'log2'</code> = "log2", <code>'-log2'</code> = "-log2", <code>'log10'</code> = "log10", <code>'-log10'</code> = "-log10", <code>'Z score'</code> = "zscore", <code>'regularized log'</code> = "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

and, 3  
andUI, 4

categoricalPalettes, 5  
Clarion, 5, 20, 22, 24, 26, 31, 35, 37, 39  
colorPicker, 8  
colorPickerUI, 9  
columnSelector, 10  
columnSelectorUI, 11  
create.logger, 42  
create\_geneview, 12  
create\_heatmap, 13  
create\_pca, 14  
create\_scatterplot, 16

divergingPalettes, 17  
download, 18  
downloadHandler, 18

equalize, 19

featureSelector, 19  
featureSelectorGuide, 20  
featureSelectorUI, 21  
force, 21  
forceAndCall, 21  
forceArgs, 21  
fread, 35

geneView, 22  
geneViewGuide, 23  
geneViewUI, 23  
geom\_point, 16  
geom\_text, 16  
get, 21  
global\_cor\_heatmap, 24  
global\_cor\_heatmap\_guide, 25  
global\_cor\_heatmapUI, 25  
grep, 43

heatmap, 26

heatmapGuide, 27  
heatmapr, 14  
heatmapUI, 27

install\_app, 28

label, 28  
labelUI, 29  
levellog, 31  
limit, 30  
limitUI, 30  
lims, 17  
log\_message, 31

make.unique, 29  
marker, 31, 39  
markerUI, 32  
match.call, 21

onFlush, 42  
orNumeric, 32  
orNumericUI, 33  
orTextual, 34  
orTextualUI, 35

parse\_MaxQuant, 36  
parser, 35  
pca, 37  
pcaGuide, 38  
pcaUI, 38  
points, 15

release\_questions, 39

scatterPlot, 39  
scatterPlotGuide, 40  
scatterPlotUI, 41  
searchData, 41  
selectInput, 34, 35, 45  
sequentialPalettes, 42  
set\_logger, 42

sliderInput, [33](#)

tag, [4](#), [10](#), [11](#), [21](#), [23](#), [25](#), [27](#), [30–33](#), [35](#), [38](#),  
[41](#), [45](#)

tobias\_parser, [43](#)

toJSON, [18](#)

transformation, [44](#)

transformationUI, [45](#)

zipr, [18](#)