Package ‘tidyHeatmap’

February 23, 2020

Type Package
Title A Tidy Implementation of Heatmap
Version 0.99.9
Maintainer Stefano Mangiola <mangiolastefano@gmail.com>
Description This is a tidy implementation for heatmap.
At the moment it is based on the (great) package 'ComplexHeatmap'.
The goal of this package is to interface a tidy data frame with this powerful tool.
Some of the advantages are:
Row and/or columns colour annotations are easy to integrate just specifying one parameter (column names).
Custom grouping of rows is easy to specify providing a grouped tbl. For example: df %>% group_by(...).
Labels size adjusted by row and column total number.
Default use of Brewer and Viridis palettes.
License GPL-3
Encoding UTF-8
LazyData true
VignetteBuilder knitr
RoxygenNote 7.0.2
biocViews AssayDomain, Infrastructure
Biarch true
Depends R (>= 3.6), ComplexHeatmap
Imports dplyr, magrittr, tidyr, rlang, purrr, stats, viridis, circlize, RColorBrewer, grid, grDevices, utils
Suggests testthat, BiocManager, knitr, qpdf, covr
URL https://www.r-project.org,
https://github.com/stemangiola/tidyHeatmap
BugReports https://github.com/stemangiola/tidyHeatmap
NeedsCompilation no
Author Stefano Mangiola [aut, cre]
add_attr

Add attribute to abject

Description

Add attribute to abject

Usage

add_attr(var, attribute, name)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>var</td>
<td>A tibble</td>
</tr>
<tr>
<td>attribute</td>
<td>An object</td>
</tr>
<tr>
<td>name</td>
<td>A character name of the attribute</td>
</tr>
</tbody>
</table>
**add_class**

**Value**
A tibble with an additional attribute

**as_matrix**

**Description**
Get matrix from tibble

**Usage**
as_matrix(tbl, rownames = NULL, do_check = TRUE)

**Arguments**
tbl A tibble
rownames A character string of the rownames
do_check A boolean

**Value**
A matrix
drop_class  Remove class to object

Description
Remove class to object

Usage
drop_class(var, name)

Arguments
var  A tibble
name  A character name of the class

Value
A tibble with an additional attribute

error_if_log_transformed

Check whether a numeric vector has been log transformed

Description
Check whether a numeric vector has been log transformed

Usage
error_if_log_transformed(x, .abundance)

Arguments
x  A numeric vector
.abundance  A character name of the transcript/gene abundance column

Value
NA
get_abundance_norm_if_exists

Get column names either from user or from attributes

Description
Get column names either from user or from attributes

Usage
get_abundance_norm_if_exists(.data, .abundance)

Arguments
- .data A tibble
- .abundance A character name of the abundance column

Value
A list of column enquo or error

get_elements

Get column names either from user or from attributes

Description
Get column names either from user or from attributes

Usage
get_elements(.data, .element, of_samples = TRUE)

Arguments
- .data A tibble
- .element A character name of the sample column
- of_samples A boolean

Value
A list of column enquo or error
get_elements_features  Get column names either from user or from attributes

Description
Get column names either from user or from attributes

Usage
get_elements_features(.data, .element, .feature, of_samples = TRUE)

Arguments
- .data: A tibble
- .element: A character name of the sample column
- .feature: A character name of the transcript/gene column
- of_samples: A boolean

Value
A list of column enquo or error

get_elements_features_abundance  Get column names either from user or from attributes

Description
Get column names either from user or from attributes

Usage
get_elements_features_abundance(
  .data,
  .element,
  .feature,
  .abundance,
  of_samples = TRUE
)
**get_sample_counts**

**Arguments**

- **.data** A tibble
- **.element** A character name of the sample column
- **.feature** A character name of the transcript/gene column
- **.abundance** A character name of the read count column
- **of_samples** A boolean

**Value**

A list of column enquo or error

---

**get_sample_counts**  
*Get column names either from user or from attributes*

**Description**

Get column names either from user or from attributes

**Usage**

```
get_sample_counts(.data, .sample, .abundance)
```

**Arguments**

- **.data** A tibble
- **.sample** A character name of the sample column
- **.abundance** A character name of the read count column

**Value**

A list of column enquo or error
**get_sample_transcript**  
*Get column names either from user or from attributes*

**Description**
Get column names either from user or from attributes

**Usage**
```
get_sample_transcript(.data, .sample, .transcript)
```

**Arguments**
- `.data` A tibble
- `.sample` A character name of the sample column
- `.transcript` A character name of the transcript/gene column

**Value**
A list of column enquo or error

---

**get_sample_transcript_counts**  
*Get column names either from user or from attributes*

**Description**
Get column names either from user or from attributes

**Usage**
```
get_sample_transcript_counts(.data, .sample, .transcript, .abundance)
```

**Arguments**
- `.data` A tibble
- `.sample` A character name of the sample column
- `.transcript` A character name of the transcript/gene column
- `.abundance` A character name of the read count column

**Value**
A list of column enquo or error
**Description**

get_x_y_annotation_columns

**Usage**

get_x_y_annotation_columns(.data, .horizontal, .vertical, .abundance)

**Arguments**

- **.data**
  A `tbl` formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
- **.horizontal**
  The name of the column horizontally presented in the heatmap
- **.vertical**
  The name of the column vertically presented in the heatmap
- **.abundance**
  The name of the transcript/gene abundance column

**Value**

A list

---

**heatmap**

*Creates a* `ComplexHeatmap` *plot from* `tbl`

**Description**

plot_heatmap() takes a tbl object and easily produces a ComplexHeatmap plot, with integration with tibble and dplyr frameworks.

**Usage**

```r
heatmap(
  .data, 
  .horizontal, 
  .vertical, 
  .abundance, 
  annotation = NULL, 
  log_transform = FALSE 
)
```
Arguments

.data A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.horizontal The name of the column horizontally presented in the heatmap
.vertical The name of the column vertically presented in the heatmap
.abundance The name of the transcript/gene abundance column
.annotation Vector of quotes
.log_transform A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)

Details

To be added.

Value

A ‘ComplexHeatmap’ object

Examples

```r
library(dplyr)
tidyHeatmap::N52 %>%
group_by(`Cell type`) %>%
tidyHeatmap::heatmap(
  .horizontal = UBR,
  .vertical = symbol_ct,
  .abundance = `read count normalised log`,
  .annotation = CAPRA_TOTAL
)
```

ifelse2_pipe

This is a generalisation of ifelse that accepts an object and return an objects

Description

This is a generalisation of ifelse that accepts an object and return an objects

Usage

```r
ifelse2_pipe(.x, .p1, .p2, .f1, .f2, .f3 = NULL)
```
ifelse_pipe

Arguments

- **.x**  
  A tibble
- **.p1**  
  A boolean
- **.p2**  
  ELSE IF condition
- **.f1**  
  A function
- **.f2**  
  A function
- **.f3**  
  A function

Value

A tibble

---

*This is a generalisation of ifelse that accepts an object and return an objects*

Description

This is a generalisation of ifelse that accepts an object and return an objects

Usage

```r
ifelse_pipe(.x, .p, .f1, .f2 = NULL)
```

Arguments

- **.x**  
  A tibble
- **.p**  
  A boolean
- **.f1**  
  A function
- **.f2**  
  A function

Value

A tibble
Example data set N52

Description

Example data set N52

Usage

N52

Format

An object of class tbl_df (inherits from tbl.data.frame) with 520 rows and 15 columns.

parse_formula

Description

.formula parser

Usage

parse_formula(fm)

Arguments

fm a formula

Value

A character vector
### pasilla

**Example data set Pasilla**

#### Description

Example data set Pasilla

#### Usage

```
# pasilla
```

#### Format

An object of class `tbl_df` (inherits from `tbl, data.frame`) with 504 rows and 6 columns.

### plot_heatmap

**plot_heatmap**

#### Description

`plot_heatmap()` takes a tbl object and easily produces a ComplexHeatmap plot, with integration with tibble and dplyr frameworks.

#### Usage

```
plot_heatmap(
  .data,  
  .horizontal,  
  .vertical,  
  .abundance,  
  annotation = NULL,  
  log_transform = FALSE 
)
```

#### Arguments

- **.data**
  A `tbl` formatted as `| <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |`
- **.horizontal**
  The name of the column horizontally presented in the heatmap
- **.vertical**
  The name of the column vertically presented in the heatmap
- **.abundance**
  The name of the transcript/gene abundance column
- **annotation**
  Vector of quotes
- **log_transform**
  A boolean, whether the value should be log-transformed (e.g., `TRUE` for RNA sequencing data)
Details
To be added.

Value
A ‘ComplexHeatmap’ object

prepend
From rlang deprecated

Description
From rlang deprecated

Usage
prepend(x, values, before = 1)

Arguments
x An array
values An array
before A boolean

Value
An array

scale_design
Scale design matrix

Description
Scale design matrix

Usage
scale_design(df, .formula)

Arguments
df A tibble
.formula a formula

Value
A tibble
select_closest_pairs

Sub function of remove_redundancy_elements_though_reduced_dimensions

Description
Sub function of remove_redundancy_elements_though_reduced_dimensions

Usage
select_closest_pairs(df)

Arguments
df A tibble

Value
A tibble with pairs to drop
Index

*Topic **datasets**
  N52, 12
  pasilla, 13

add_attr, 2
add_class, 3
as_matrix, 3

drop_class, 4

error_if_log_transformed, 4

get_abundance_norm_if_exists, 5
get_elements, 5
get_elements_features, 6
get_elements_features_abundance, 6
get_sample_counts, 7
get_sample_transcript, 8
get_sample_transcript_counts, 8
get_x_y_annotation_columns, 9

heatmap, 9

ifelse2_pipe, 10
ifelse_pipe, 11

N52, 12

parse_formula, 12
pasilla, 13
plot_heatmap, 13
prepend, 14

scale_design, 14
select_closest_pairs, 15