Package 'TableContainer'

May 7, 2026

Title Create a Table with Row, Column, and Table Annotations

Version 1.0.0

Description Offers a TableContainer() function to create tables enriched with row, column, and table annotations. This package is similar to 'SummarizedExperiment' in Bioconductor <doi:10.18129/B9.bioc.SummarizedExperiment>, but designed to work independently of Bioconductor, it ensures annotations are automatically updated when the table is subset. Additionally, it includes format_tbl() methods for enhanced table formatting and display.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Biarch true

Depends R (>= 3.5)

Imports methods, cli, glue, utils

SystemRequirements GNU make

Suggests knitr, rmarkdown, testthat (>= 3.2.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Jiefei Wang [aut, cre] (ORCID: <https://orcid.org/0000-0002-2709-5332>)

Maintainer Jiefei Wang <szwjf08@gmail.com>

Repository CRAN

Date/Publication 2025-06-19 07:30:02 UTC

Contents

`.printTable` *Show method for TableContainer and its components*

Description

The show method uses `.printTable` to display the table and `.printMeta` to display the rowMeta, colMeta, and metaData.

Usage

```
.printTable(tbl, ...)

.printMeta(meta, name)

## S4 method for signature 'TableContainer'
show(object)
```

Arguments

`tbl`	A table-like object (e.g., matrix, data.frame).
`...`	Additioanl arguments passed to the `format_tbl` function.
`meta`	A list of metadata items.
`name`	A string representing the name of the metadata.
`object`	A TableContainer object.

Value

`.printTable`: No return value, called for side effects of printing the table to the console.

`.printMeta`: No return value, called for side effects of printing metadata information to the console.

`show`: No return value, called for side effects of displaying the TableContainer object contents to the console.

`as.matrix,TableContainer-method`
Data Conversion

Description

Data Conversion

Usage

```
## S4 method for signature 'TableContainer'
as.matrix(x)

## S4 method for signature 'TableContainer'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

x	A TableContainer object.
row.names	NULL or a character vector giving the row names for the data frame. Missing values are not allowed. See base::data.frame for more details.
optional	Logical. If TRUE, setting row names is optional. See base::data.frame for more details.
...	additional arguments

Value

as.matrix: A matrix representation of the TableContainer object.

as.data.frame: A data frame representation of the TableContainer object.

---

dim,TableContainer-method

Container Methods

Description

Container Methods

Usage

```
## S4 method for signature 'TableContainer'
dim(x)

## S4 method for signature 'TableContainer'
dimnames(x)

## S4 method for signature 'TableContainer'
nrow(x)

## S4 method for signature 'TableContainer'
ncol(x)

## S4 method for signature 'TableContainer'
rownames(x)
```

```
## S4 replacement method for signature 'TableContainer'
rownames(x) <- value

## S4 method for signature 'TableContainer'
colnames(x)

## S4 replacement method for signature 'TableContainer'
colnames(x) <- value

## S4 method for signature 'TableContainer'
x[i, j, ..., drop = TRUE]

tblData(object)

tblData(object) <- value

rowData(object)

rowData(object) <- value

colData(object)

colData(object) <- value

metaData(object)

metaData(object) <- value

## S4 method for signature 'TableContainer'
tblData(object)

## S4 replacement method for signature 'TableContainer'
tblData(object) <- value

## S4 method for signature 'TableContainer'
rowData(object)

## S4 replacement method for signature 'TableContainer'
rowData(object) <- value

## S4 method for signature 'TableContainer'
colData(object)

## S4 replacement method for signature 'TableContainer'
colData(object) <- value

## S4 method for signature 'TableContainer'
metaData(object)
```

```
## S4 replacement method for signature 'TableContainer'
metaData(object) <- value
```

Arguments

x	A TableContainer object.
value	A matrix, data.frame, or NULL.
i	Row indices for subsetting. If only i is provided, it will return the entire row(s).
j	Column indices for subsetting.
...	Additional arguments.
drop	Not used.
object	A TableContainer object.

Value

dim, dimnames, nrow, and ncol: the respective dimensions, dimnames, number of rows, and number of columns of the TableContainer object. When the table slot is NULL, the dimensions are derived from the rowData and colData slots.

[: A new TableContainer object with the selected data.

tblData, rowData, colData, and metadata: the respective slots of the TableContainer object.
tblData<-, rowData<-, colData<-, and metadata<-: update the respective slots and return the modified object.

---

format_tbl *Format a table for display. The maximum number of columns displayed is limited by the terminal width.*

Description

Format a table for display. The maximum number of columns displayed is limited by the terminal width.

Usage

```
format_tbl(
  tbl,
  max_tbl_width,
  soft_table_width = max_tbl_width,
  max_row = 8,
  cell_formatter = common_formatter,
  delimiter = " ",
  truncate_marker = " ...",
  header_name = "",
  leading_text = NULL,
```

```
  include_row_names = TRUE,
  include_col_names = TRUE
)
```

Arguments

tbl	A table-like object (e.g., matrix, data.frame).
max_tbl_width	The maximum width for the table display.
soft_table_width	A softer width limit for the table display, allowing the last column to exceed it.
max_row	The maximum number of rows to display.
cell_formatter	A function to format individual cells.
delimiter	The delimiter to use for separating columns.
truncate_marker	A marker to indicate truncation of the table.
header_name	A string to print as the first element in the column names row. It only works when include_col_names = TRUE.
leading_text	A string to print before the table. This can be used to indent the entire table. It can be a single string or a vector of strings. If it is a vector, it will be recycled to match the number of rows in the table.
include_row_names	Logical, whether to include row names in the display.
include_col_names	Logical, whether to include column names in the display.

Value

A formatted string vector representing the table. The length of the vector is the number of rows in the table.

See Also

common_formatter() for the cell formatting function and pretty_number() for formatting numeric values.

Examples

```
## Format the table
tbl <- data.frame(
    x = c(1, 123, 123456678, 1235678887644),
   y = c("abc", "this is a long string", "another long string", "yet another long string"),
    z = c(TRUE, FALSE, TRUE, FALSE),
    d = runif(4) * 100000
)
formatted_tbl <- format_tbl(tbl, max_tbl_width = 50, max_row = 3)
cat(formatted_tbl, sep = "\n")
```

pretty_number *Common functions for formatting table cells*

Description

Common functions for formatting table cells

Common functions for formatting table cells

Usage

```
pretty_number(x, max_len = 20)

common_formatter(x, max_len = 20)
```

Arguments

x	A vector of values to format.
max_len	The maximum length for the formatted string.

Details

`pretty_number` : keep all digits if we can keep it within the width limit. Otherwise, use scientific notation to reduce length. If the number still cannot fit within the length limit, return the shortest result.

`common_formatter` : For numeric, call `pretty_number` to format the number. For non-numeric, truncate the string and append "..." if it exceeds the width limit.

Value

A character string with the formatted number. If the number cannot fit within the specified maximum length, scientific notation is used with reduced digits.

A character vector of the same length as x with formatted values. For numeric values, uses pretty_number() formatting. For non-numeric values, truncates strings longer than max_len and appends "..." if needed.

See Also

`format_tbl()` for formatting tables

Examples

```
## Format a number
pretty_number(1234567890, max_len = 20)
pretty_number(1234567890, max_len = 8)
pretty_number(1234567890, max_len = 3)
## format character
common_formatter("this is a long string", max_len = 40)
common_formatter("this is a long string", max_len = 20)
```

TableContainer *The constructor function for TableContainer*

Description

Creates a TableContainer object with the specified matrix, rowData, colData, and metadata.

Usage

```
TableContainer(table = NULL, rowData = NULL, colData = NULL, metaData = NULL)
```

Arguments

table	A matrix, data.frame, or NULL.
rowData	A data.frame or NULL object describing the rows.
colData	A data.frame or NULL object describing the columns.
metaData	A list or NULL containing arbitrary metadata associated with the overall data.

Value

A TableContainer object.

Examples

```
tbl <- matrix(1:12, nrow = 3, ncol = 4)
row_dt <- data.frame(row1 = 1:3, row2 = letters[1:3])
col_dt <- data.frame(col1 = 1:4, col2 = letters[1:4])
meta_dt <- list(meta1 = "meta1", meta2 = "meta2")

TableContainer(
    table = tbl,
    rowData = row_dt,
    colData = col_dt,
    metaData = meta_dt
)
```

TableContainer-class *The TableContainer Class*

Description

A container for a matrix and associated row/column annotations. This is similar in concept to Bioconductor's SummarizedExperiment but with no Bioconductor dependencies.

Slots

`table` A matrix, data.frame, or NULL.

`rowData` A data.frame or NULL object describing the rows. Each row of the matrix corresponds to a row in the rowData object.

`colData` A data.frame or NULL object describing the columns. Each column of the matrix corresponds to a row in the colData object.

`metaData` A list or NULL containing arbitrary metadata associated with the overall data.

Index