

Package: gplate (via r-universe)

November 3, 2024

Title A Grammar of Plates

Version 0.2.0

Description `gplate` attempts to provide a succinct yet powerful grammar to describe common microwell layouts to aide in both plotting and tidying.

License GPL (>= 3)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports cli, dplyr, ggplot2, glue, methods, purrr, rlang, stats, tibble, tidyr

Suggests rmarkdown, knitr, testthat (>= 3.0.0)

VignetteBuilder knitr

Depends R (>= 4.1.0)

URL <https://kaiaragaki.github.io/gp/>

Config/testthat/edition 3

Repository <https://kaiaragaki.r-universe.dev>

RemoteUrl <https://github.com/KaiAragaki/gplate>

RemoteRef HEAD

RemoteSha 7d5c8935eee7c618cf44477fee82ad2fc864ed38

Contents

arrange_by_dim	2
as_gp	3
coordinate	3
example_plate	4
flip_dim	4
gp	5

gp_excise	6
gp_mini_theme	7
gp_plot	7
gp_sec	8
gp_serve	9
gp_unravel	10
gp_unserve	10
is_fwd	11
new_gp	12
non_int_replicate	12
pcr_plate	13
plate_formats	13
protein_quant	13
rel_dim	14
unroll_sec_dim_along_parent	14
wells	15
well_data	15
Index	16

arrange_by_dim	<i>Arrange gp well data relative to user supplied start_corner</i>
----------------	--

Description

Arrange gp well data relative to user supplied start_corner

Usage

```
arrange_by_dim(gp, dim)
```

Arguments

gp	A gp object
dim	Character. Either "row" or "col"

Value

A gp with gp\$well_data arranged by dim and the non-dim (ie if dim = "row", the 'non-dim' is "col") relative to the start corner (start corner is at the top, farther from start corner is at bottom)

as_gp	<i>Coerce object to gp</i>
-------	----------------------------

Description

Coerce object to gp

Usage

```
as_gp(x, ...)
```

```
## Default S3 method:
as_gp(x, ...)
```

```
## S3 method for class 'list'
as_gp(x, ...)
```

Arguments

x	Object to coerce
...	Unused

coordinate	<i>Create a dimension pattern 'unit' of a section</i>
------------	---

Description

Create a dimension pattern 'unit' of a section

Usage

```
coordinate(gp, type = c("row", "col"), margin)
```

Arguments

gp	a gp
type	Character. Either 'row' or 'col'.
margin	Named list of integers defining the size of margins at each edge.

Value

a tibble

example_plate	<i>A 8x12 matrix of random integers 1-10</i>
---------------	--

Description

A 8x12 matrix of random integers 1-10

Usage

```
example_plate
```

Format

An object of class `matrix` (inherits from `array`) with 8 rows and 12 columns.

flip_dim	<i>Take a set of numbers and flip them around on a number line</i>
----------	--

Description

Take a set of numbers and flip them around on a number line

Usage

```
flip_dim(gp, dim)
```

Arguments

gp	A gp
dim	Symbol. Column to flip. Should be the name of a column that exists in <code>gp\$well_data</code>

Details

This assumes the beginning is 1 and the end is maximum length of the dimension. Note that this does NOT simply check the maximum value of the supplied vector, but searches the supplied gp for a matching `ndim` column that definitively tells how long the given dimension is supposed to be. This is important, as sometimes a dimension is given that is longer than its parent dimension, so the maximum numbers would not show in the vector.

Value

A vector.

gp *Make a gp object*

Description

Make a gp object

Usage

```
gp(rows = NULL, cols = NULL, data = NULL, wells = NULL, tidy = FALSE)
```

Arguments

rows	Numeric. The number of rows the plate should have.
cols	Numeric. The number of columns the plate should have.
data	An optional data.frame of well data the same dimensions as the plate to be described
wells	Numeric. The number of wells the plate has. If this is specified, rows and cols must be null - they are inferred from common form factors of plates.
tidy	Are the data supplied tidy?

Details

A gp object has the following components:

- `nrow/ncol`: Number of plate rows/cols. This is static and will not be changed by adding layers.
- `well_data`: Somewhat transient data used to define plotting coordinates for layers. See below for more information.
- `nrow_sec/ncols_sec`: The number of rows/cols of the current section. When creating a plate, that number is the number of rows/cols of the plate (the plate is the section). Can take on a 'mar' suffix, which specifies the number including margins (if any)
- `nrow_sec_par/ncol_sec_par`: The number of rows/cols of the parent section. When creating a plate, it has no parent, so defaults to being its own parent. Can take on a 'mar' suffix. See above.

`well_data` consists of many columns. The variable names can be broken down as follows:

- `row/col`: row is always the y axis, col is always the x axis. By convention, plates start at 1, 1 in the top left corner.
- `sec`: Short for 'section'. A section is a rectangular field of wells. `sec` alone refers to the number of the section itself. `sec` combined with `row` or `col` (eg `row_sec`) refers to the coordinates of a given well relative to it's section corners, with the top left corner of a given section always being (1, 1).
- `par`: Short for 'parent'. These columns are all the data from the previous layer.
- `index`: These specify (usually) multiwell strips only defined in one dimension. Together, `index_row` and `index_col` form checkerboard-like patterns, where each intersection is a section. This is a bit more complicated when `wrap = TRUE`, so the simile does not hold for all cases.

Value

a gp object

Examples

```
# If you specify wells, rows and columns are derived
# from a standard plate sizes:
```

```
gp(wells = 96)
```

```
# As such, you cannot use the wells argument
# if you want to create more exotic plates:
```

```
try(gp(wells = 102))
```

```
# For that, you'll need to specify wells and cols:
```

```
gp(rows = 6, cols = 17)
```

gp_excise

Slice out a smaller gp from a gp via coordinates

Description

Slice out a smaller gp from a gp via coordinates

Usage

```
gp_excise(gp, top, left, bottom, right)
```

Arguments

gp A gp

top, left, bottom, right

Integer. The coordinates of the corners. Remember that TOP left is (1, 1)

Value

A gp with top left coords = (1, 1)

Examples

```
gplate::example_plate |> as_gp() |> gp_excise(2, 2, 4, 4)
```

gp_mini_theme	<i>A theme for making little in-line plots</i>
---------------	--

Description

A theme for making little in-line plots

Usage

```
gp_mini_theme()
```

Value

A ggplot2 theme

gp_plot	<i>Plot a gp object</i>
---------	-------------------------

Description

Plot a gp object

Usage

```
gp_plot(x, name = .sec, ...)  
  
## S3 method for class 'gp'  
gp_plot(x, name = .sec, ...)  
  
## S3 method for class 'data.frame'  
gp_plot(x, name = .sec, ...)
```

Arguments

x	A gp object or data.frame
name	Symbol. Name of a column in gp\$well_data (or a column in the data.frame if data.frame was supplied) to use as a color.
...	Additional arguments to be passed to ggplot2::geom_point()

Value

a ggplot

Examples

```
gp(16, 24) |> gp_plot(.row)
```

gp_sec

Add a section to a gp object.

Description

Add a section to a gp object.

Usage

```
gp_sec(gp, ...)

## S3 method for class 'gp'
gp_sec(
  gp,
  name,
  nrow = NULL,
  ncol = NULL,
  labels = NULL,
  start_corner = c("tl", "tr", "bl", "br"),
  flow = c("row", "col"),
  margin = 0,
  wrap = FALSE,
  break_sections = TRUE,
  advance = TRUE
)
```

Arguments

gp	A gp object
name	Character. Name of the section.
nrow	Numeric. Section height. If NULL, will fill width of parent section.
ncol	Numeric. Section width. If NULL, will fill width of parent section.
labels	Optional. What should the labels of each section be?
start_corner	Corner of section to place first item.
flow	Direction that subsequent items should be placed relative to first corner.
margin	Border width outside the section that will be unfilled. Can take an argument of one (same border all around), two (top/bottom, left/right), three (top, left/right, bottom), or four (top, right, bottom, left).
wrap	Should the sections that go off the edge continue on the next row/column?
break_sections	Should partial sections be allowed?
advance	Should this section be a child or sibling of the one before it? If TRUE (default), it will be a child.

Value

a gp

Examples

```
gp(16, 24) |> gp_sec("section 1", ncol = 3)

pq <- gp(8, 12, protein_quant) |> gp_sec("has_sample", 3, 19, wrap = TRUE, labels = "sample")

# Sections can be used to label things for tidying
pq |> gp_serve()

# They can also be used for plotting:
pq |> gp_plot(has_sample)
```

gp_serve

Extract useful, tidy data from a gp object

Description

Extract useful, tidy data from a gp object

Usage

```
gp_serve(gp)
```

Arguments

gp a gp

Value

a tibble, with .row and .col, as well as any created section names and data values.

Examples

```
gp(16, 24) |>
  gp_sec("my_sec", nrow = 9, ncol = 7, labels = c("sample_1", "sample_2", "sample_3")) |>
  gp_serve()
```

gp_unravel

Turn data from 'plate form' to 'tidy form' and back

Description

Turn data from 'plate form' to 'tidy form' and back

Usage

```
gp_unravel(df, rownames = NULL)
```

```
gp_reravel(df, row_name = ".row", col_name = ".col", values = "value")
```

Arguments

df The data.frame in plate-form to be tidied

rownames Optional character. If there is a colname that specifies the row index, it will be arranged by this column, then dropped.

Value

a tibble

Examples

```
test_plate <- matrix(sample(1:10, 96, replace = TRUE), nrow = 8, ncol = 12)
```

```
gp_unravel(test_plate)
```

gp_unserve

Turn a tidy data.frame into a gp

Description

An opposite to serve()

Usage

```
gp_unserve(x, row = ".row", col = ".col", nrow = NULL, ncol = NULL)
```

Arguments

x	A tidy data . frame representation of plate data
row	Character. The column that represents the rows. Should be numeric, with the lowest number representing the topmost row.
col	Character. The name of the column that represents columns. Should be numeric, with the lowest number representing the leftmost column.
nrow	Optional numeric, denoting the number of rows in the plate. If not supplied, will be imputed from the largest number in the 'row' column
ncol	Optional numeric, denoting the number of columns in the plate. If not supplied, will be imputed from the largest number in the 'col' column

Value

a gp

is_fwd

Check if axis moves in the canonical direction

Description

'Forwards' is thought of 'left to right' when thinking about moving across columns and 'top to bottom' when moving across rows

Usage

```
is_fwd(gp, dim)
```

Arguments

gp	A gp
dim	Character. A dimension, either "row" or "col".

Value

logical.

new_gp *A constructor for a gp object*

Description

A constructor for a gp object

Usage

```
new_gp(nrow = 1L, ncol = 1L, data = data.frame(), tidy = FALSE)
```

Arguments

nrow	Integer. Number of rows of the plate.
ncol	Integer. Number of columns of the plate.
data	An optional dataframe including plate data
tidy	Is the supplied data already tidy, or should it be tidied?

Value

a gp object

Examples

```
new_gp(nrow = 8L, ncol = 16L)
```

non_int_replicate *Recycle a df a non-integer number of times*

Description

Recycle a df a non-integer number of times

Usage

```
non_int_replicate(df, measure)
```

Arguments

df	data.frame to be replicated
measure	Length-out of replication (not number of times). Can be a numeric, a vector, or a df. If a df, will use number of rows. If a vector, will use length.

Value

a data.frame with nrow measure

pcr_plate	<i>Data from a PCR experiment on a 384-well plate</i>
-----------	---

Description

Data from a PCR experiment on a 384-well plate

Usage

pcr_plate

Format

An object of class `matrix` (inherits from `array`) with 16 rows and 24 columns.

plate_formats	<i>Standard plate formats for microwell plates</i>
---------------	--

Description

Standard plate formats for microwell plates

Usage

plate_formats

Format

An object of class `data.frame` with 9 rows and 3 columns.

protein_quant	<i>Data from a protein quantification assay in a 96-well plate. See the "Using gp to wrangle plate data" vignette for more details</i>
---------------	--

Description

Data from a protein quantification assay in a 96-well plate. See the "Using gp to wrangle plate data" vignette for more details

Usage

protein_quant

Format

An object of class `matrix` (inherits from `array`) with 8 rows and 12 columns.

rel_dim	<i>Performs flip_dim if necessary</i>
---------	---------------------------------------

Description

Performs flip_dim if necessary

Usage

```
rel_dim(gp, dim, rel)
```

Arguments

gp	A gp
dim	Symbol. Column to conditionally flip. Should be the name of a column that exists in gp\$well_data
rel	Character. Column to check if_fwd on. If TRUE, return dim as is. If FALSE (rel is backwards), flip the column.

Value

A vector that is flipped (see flip_dim) if rel is backwards (see is_fwd)

unroll_sec_dim_along_parent	<i>Repeat one dimension of a child section across a dimension of a parent section</i>
-----------------------------	---

Description

Repeat one dimension of a child section across a dimension of a parent section

Usage

```
unroll_sec_dim_along_parent(gp, dim, wrap)
```

Arguments

gp	A gp
dim	Either "row" or "col"
wrap	Logical. Should this dimension wrap around when it hits a section boundary?

Value

A gp

wells	<i>Calculate wells of gp object</i>
-------	-------------------------------------

Description

Calculate wells of gp object

Usage

```
wells(x, ...)  
  
## S3 method for class 'gp'  
wells(x, ...)
```

Arguments

x	a gp
...	Unused

well_data	<i>Extract well data from an object</i>
-----------	---

Description

Extract well data from an object

Usage

```
well_data(x, ...)  
  
## S3 method for class 'gp'  
well_data(x, ...)
```

Arguments

x	An object to be passed to its respective method
...	Additional arguments (unused)

Value

A tibble containing the well data of the object

Index

* datasets

- example_plate, 4
- pcr_plate, 13
- plate_formats, 13
- protein_quant, 13

arrange_by_dim, 2

as_gp, 3

coordinate, 3

example_plate, 4

flip_dim, 4

gp, 5

gp_excise, 6

gp_mini_theme, 7

gp_plot, 7

gp_reravel (gp_unravel), 10

gp_sec, 8

gp_serve, 9

gp_unravel, 10

gp_unserve, 10

is_fwd, 11

new_gp, 12

non_int_replicate, 12

pcr_plate, 13

plate_formats, 13

protein_quant, 13

rel_dim, 14

unroll_sec_dim_along_parent, 14

well_data, 15

wells, 15