

# Package ‘platetools’

April 24, 2019

**Title** Tools and Plots for Multi-Well Plates

**Version** 0.1.2

**Description** Collection of functions for working with multi-well microtitre plates, mainly 96, 384 and 1536 well plates.

**Depends** R (>= 3.1.0)

**License** MIT + file LICENSE

**URL** <https://github.com/swarchal/platetools>

**BugReports** <https://github.com/swarchal/platetools/issues>

**Encoding** UTF-8

**LazyData** true

**Imports** RColorBrewer, ggplot2 (>= 2.2.0)

**Suggests** testthat, viridis

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Author** Scott Warchal [aut, cre]

**Maintainer** Scott Warchal <[scott.warchal@gmail.com](mailto:scott.warchal@gmail.com)>

**Repository** CRAN

**Date/Publication** 2019-04-24 07:20:03 UTC

## R topics documented:

bhit_map . . . . .	2
b_grid . . . . .	3
b_map . . . . .	4
b_score . . . . .	5
check_plate_input . . . . .	6
dist_map . . . . .	6
fill_plate . . . . .	7
hit_grid . . . . .	7
hit_map . . . . .	8

is_1536 . . . . .	9
list_to_dataframe . . . . .	10
med_smooth . . . . .	10
missing_wells . . . . .	11
num_to_well . . . . .	11
pchit_grid . . . . .	12
pchit_map . . . . .	13
pc_grid . . . . .	14
pc_map . . . . .	15
plate_effect . . . . .	15
plate_map . . . . .	16
plate_map_grid . . . . .	16
plate_map_grid_scale . . . . .	17
plate_map_multiple . . . . .	17
plate_map_scale . . . . .	18
plate_matrix . . . . .	18
plt1536 . . . . .	19
plt384 . . . . .	19
plt96 . . . . .	20
raw_grid . . . . .	20
raw_map . . . . .	21
readmap_data . . . . .	22
read_map . . . . .	22
rotate_plate . . . . .	23
set_block . . . . .	23
well_to_num . . . . .	24
z_grid . . . . .	25
z_map . . . . .	26

<b>Index</b>	<b>28</b>
--------------	-----------

---

bhit_map	<i>Platemap to identify 'hits' following a B-score normalisation</i>
----------	--

---

## Description

Produces a platemap with colours indicating wells above or below selected threshold after normalising for systematic plate effects via B-score smooth. The threshold is defined calculated from a z-score, i.e plus or minus standard deviations from the plate mean.

## Usage

```
bhit_map(data, well, plate = 96, threshold = 2, palette = "Spectral",
          eps = 0.01, maxiter = 10, trace.iter = FALSE, na.rm = TRUE, ...)
```

**Arguments**

data	Vector of numerical values
well	Vector of well identifiers, e.g "A01"
plate	Number of wells in whole plate (96, 384 or 1536)
threshold	Standard deviations from the plate average to indicate a hit. default is set to +/- 2 SD.
palette	RColorBrewer palette
eps	real number greater than 0. A tolerance for divergence
maxiter	int, the maximum number of iterations
trace.iter	Boolean, should progress in convergence be reported?
na.rm	Boolean, should missing values be removed?
...	additional parameters to plot wrappers

**Value**

ggplot plot

**Examples**

```
df <- data.frame(vals = rnorm(384),
  well = num_to_well(1:384, plate = 384))

bhit_map(data = df$vals,
  well = df$well,
  plate = 384,
  threshold = 3)
```

---

b\_grid

*Plots multiple b-scored normalised platemaps*

---

**Description**

Transforms numerical values using the b-score normalisation process to account for row and column effects. Uses well and plate labels to plot the normalised values in the form of microtitre plates. Works for 96, 384 and 1536 well plates.

**Usage**

```
b_grid(data, well, plate_id, plate = 96, eps = 0.01, maxiter = 10,
  trace.iter = FALSE, na.rm = FALSE, ...)
```

**Arguments**

data	Numerical values to be plotted
well	Vector of well identifiers e.g "A01"
plate_id	Vector of plate identifiers e.g "Plate_1"
plate	Number of wells in complete plate (96, 384 or 1536)
eps	real number greater than 0. A tolerance for divergence
maxiter	int, the maximum number of iterations
trace.iter	Boolean, should progress in convergence be reported?
na.rm	Boolean, should missing values be removed?
...	additional parameters to plot wrappers

**Value**

ggplot plot

**Examples**

```
df01 <- data.frame(well = num_to_well(1:96),
                  vals = rnorm(96),
                  plate = 1)

df02 <- data.frame(well = num_to_well(1:96),
                  vals = rnorm(96),
                  plate = 2)

df <- rbind(df01, df02)

b_grid(data = df$vals,
       well = df$well,
       plate_id = df$plate,
       plate = 96)
```

---

b\_map

*Plots a heatmap of b-score normalised values in a plate layout*

---

**Description**

Transforms numerical values using the b-score normalisation process to account for row and column effects. Uses well labels to plot the normalised values in the form of a microtitre plate. Works for 96, 384 or 1536 well plates

**Usage**

```
b_map(data, well, normalise = FALSE, plate = 96, eps = 0.01,
      maxiter = 10, trace.iter = FALSE, na.rm = TRUE, ...)
```

**Arguments**

data	Numerical values in the form of a vector to be normalised
well	Vector of well identifiers, e.g "A01"
normalise	Boolean, if TRUE then the residual values will be divided by the plate median absolute deviation as per Malo et al.
plate	integer, 96, 384 or 1536
eps	real number greater than 0. A tolerance for divergence
maxiter	int, the maximum number of iterations
trace.iter	Boolean, should progress in convergence be reported?
na.rm	Boolean, should missing values be removed?
...	additional parameters to plot wrappers

**Value**

ggplot plot

**Examples**

```
df <- data.frame(well = num_to_well(1:96),
  vals = rnorm(96))

b_map(data = df$vals,
  well = df$well,
  plate = 96)

df_384 <- data.frame(
  well = num_to_well(1:384, plate = 384),
  vals = rnorm(384))

b_map(data = df_384$vals,
  well = df_384$well,
  plate = 384)
```

---

b_score	<i>2 way median polish</i>
---------	----------------------------

---

**Description**

2 way median polish to remove plate effects such as row/column/edge effects. Given a dataframe containing alpha-numeric wellIDs and numerical values, this b\_score will return a dataframe of the same structure after a two-way median smooth.

**Usage**

```
b_score(data, well, plate)
```

**Arguments**

data	numeric data, either a vector or dataframe column
well	alpha-numeric wellIDs. e.g 'A01'
plate	numeric, number of wells within a plate, either 96 or 384

**Examples**

```
df <- data.frame(well = num_to_well(1:96),
                 vals = rnorm(96))
```

```
b_score(data = df$vals,
        well = df$well,
        plate = 96)
```

---

check_plate_input	<i>checks plate input for dodgy well plate combinations</i>
-------------------	---

---

**Description**

checks plate input for dodgy well plate combinations

**Usage**

```
check_plate_input(well, plate)
```

**Arguments**

well	vector of well labels
plate	integer, number of wells in full plate

---

dist_map	<i>Plots distributions per well in a plate layout</i>
----------	---

---

**Description**

Produces distribution plots faceted in a plate-layout format.

**Usage**

```
dist_map(well, data)
```

**Arguments**

well	vector of alphanumeric wellIDs e.g 'A01'
data	numeric vector

**Value**

ggplot plot

---

fill_plate	<i>Fill in missing wells</i>
------------	------------------------------

---

**Description**

Fills in missing wells with rows of NA values. Useful for any functions that require a complete plate such as 'b\_score'.

**Usage**

```
fill_plate(df, well, plate = 96)
```

**Arguments**

df	dataframe
well	Column containing well identifiers i.e "A01"
plate	Number of wells in complete plate (96, 384 or 1536)

**Value**

dataframe

**Examples**

```
vals <- rnorm(96) ; wells <- num_to_well(1:96)
df <- data.frame(wells, vals)
df_missing <- df[-c(1:10), ]
fill_plate(df_missing, "wells")
```

---

hit_grid	<i>Plots multiple platemaps with and identifies hits</i>
----------	--

---

**Description**

Converts numerical values and well labels into 'hits' in the form of multiple plate maps. Hits are calculated as wells above or below a specified number of standard deviations from the overall average

**Usage**

```
hit_grid(data, well, plate_id, threshold = 2, ncols = 2, plate = 96,
  each = FALSE, scale_each = FALSE, palette = "Spectral", ...)
```

**Arguments**

data	Numerical values to be scaled and plotted
well	Vector of well identifiers. e.g "A01"
plate_id	Vector of plate identifiers e.g "Plate_1"
threshold	Numerical value of standard deviations from the mean for a well to be classified as a 'hit'. Default it +/- 2 SD
ncols	Number of columns in the grid of plates
plate	Number of wells in the complete plates (96, 384 or 1536)
each	boolean, allowed for backwards compatibility, scale_each is now the preferred argument name
scale_each	boolean, if true scales each plate individually, if false will scale the pooled values of data
palette	RColorBrewer palette
...	additional arguments for plot wrappers

**Value**

ggplot plot

**Examples**

```
df01 <- data.frame(well = num_to_well(1:96),
  vals = rnorm(96),
  plate = 1)

df02 <- data.frame(well = num_to_well(1:96),
  vals = rnorm(96),
  plate = 2)

df <- rbind(df01, df02)

hit_grid(data = df$vals,
  well = df$well,
  plate_id = df$plate,
  plate = 96,
  each = FALSE)
```

---

hit\_map

*Platemap to identify 'hits' in a screen*

---

**Description**

Produces a plot in the form of a micro-titre layout, with colours indicating wells above or below a nominated threshold.



**Usage**

```
hit_map(data, well, plate = 96, threshold = 2, palette = "Spectral", ...)
```

**Arguments**

data	Vector of numerical values to score
well	Vector of well identifiers e.g "A01"
plate	Number of wells in complete plate (96, 384 or 1536)
threshold	Numerical value of standard deviations from the mean for a well to be classified as a 'hit'. Default it +/- 2 SD
palette	RColorBrewer palette
...	additional parameters for plot wrappers

**Value**

ggplot plot

**Examples**

```
df <- data.frame(vals = rnorm(1:384),
                 well = num_to_well(1:384, plate = 384))

hit_map(data = df$vals,
        well = df$well,
        plate = 384,
        threshold = 3)
```

---

is\_1536

*internal 1536 plate function for plate\_map*

---

**Description**

internal 1536 plate function for plate\_map

**Usage**

```
is_1536(well)
```

**Arguments**

well	vector of alphanumeric well labels
------	------------------------------------

---

list_to_dataframe	<i>Converts list to a dataframe in a sensible way</i>
-------------------	---

---

**Description**

Given a list of dataframes with the same columns, this function will row bind them together, and if passed a col\_name argument, will produce a column containing their original element name

**Usage**

```
list_to_dataframe(l, col_name = NULL)
```

**Arguments**

l	list of dataframes to be converted into single dataframe
col_name	(optional) name of column to put element names under

**Value**

dataframe

---

med_smooth	<i>2-way median smooth</i>
------------	----------------------------

---

**Description**

Given a platemap produced by plate\_map, will return a dataframe with after values have been transformed into a matrix mirroring the plate structure and undergoing a 2-way median polish to remove row or column effects

**Usage**

```
med_smooth(platemap, plate, eps = 0.01, maxiter = 10, trace.iter = FALSE,
  na.rm = TRUE)
```

**Arguments**

platemap	dataframe produced by plate_map
plate	numeric, number of wells in plate, either 96 or 384
eps	real number greater than 0. A tolerance for divergence
maxiter	int, the maximum number of iterations
trace.iter	Boolean, should progress in convergence be reported?
na.rm	Boolean, should missing values be removed?

**Value**

A dataframe consisting of two column, wellID and polished numeric values

---

missing_wells	Returns wells that are missing from a complete plate
---------------	--

---

**Description**

Returns a vector of wells that are missing from a complete plate.

**Usage**

```
missing_wells(df, well, plate = 96)
```

**Arguments**

df	dataframe
well	Column containing well identifiers i.e "A01"
plate	Number of wells in complete plate (96 or 384)

**Value**

vector of missing wells

**Examples**

```
vals <- rnorm(96) ; wells <- num_to_well(1:96)
df <- data.frame(vals, wells)
df_missing <- df[-c(1:10), ]
missing_wells(df_missing, "wells")
```

---

num_to_well	Converts numbers to well labels
-------------	---------------------------------

---

**Description**

Converts numerical values to corresponding alpha-numeric well labels for 96, 384 or 1536 well plates.

**Usage**

```
num_to_well(x, plate = 96)
```

**Arguments**

x	Vector of numbers to be converted
plate	Number of wells in complete plate (96 or 384)

**Value**

Vector of alpha-numeric well labels

**Examples**

```
num_to_well(1:96)
num_to_well(1:96, plate = 384)

nums <- c(1:10, 20:40, 60:96)
num_to_well(nums)
```

---

pchit_grid	<i>Plots multiple heatmaps identifying hits from the first principal component</i>
------------	--

---

**Description**

Converts numerical values, well labels, and plate labels into multiple heatmaps of plates, with z-scored principal components coloured dependent on a specified threshold of standard deviations above or below the average.

**Usage**

```
pchit_grid(data, well, plate_id, ...)
```

**Arguments**

data	Numerical values, either a dataframe or a matrix
well	Vector of well identifiers e.g "A01"
plate_id	Vector of plate identifiers e.g "Plate_1"
...	additional arguments to 'platetools::hit_grid()'

**Value**

ggplot plot

**Examples**

```
df01 <- data.frame(
  well = num_to_well(1:96),
  plate = 1,
  vals1 = rnorm(1:96),
  vals2 = rnorm(1:96))

df02 <- data.frame(
  well = num_to_well(1:96),
  plate = 2,
  vals1 = rnorm(1:96),
```

```

vals2 = rnorm(1:96)

df <- rbind(df01, df02)

pchit_grid(data = df[,3:4],
           well = df$well,
           plate_id = df$plate,
           plate = 96)

```

---

pchit\_map

*Plots a heatmap identifying hits from the first principal component*


---

### Description

Converts numerical values and plate labels into a plate heatmap with z-scored principal components coloured dependent on a specified threshold of standard deviations above or below the average.

### Usage

```

pchit_map(data, well, plate = 96, threshold = 2, palette = "Spectral",
          ...)

```

### Arguments

data	Numerical values, either a dataframe or a matrix
well	Vector of well identifiers e.g "A01"
plate	Number of wells in complete plate (96, 384 or 1536)
threshold	Threshold of +/- standard deviations from the average to determine a hit
palette	RColorBrewer palette
...	additional arguments to <code>platetools::hit_map</code>

### Value

ggplot plot

### Examples

```

v1 <- rnorm(1:96)
v2 <- rnorm(1:96)
v3 <- rnorm(1:96)
wells <- num_to_well(1:96)
df <- data.frame(wells, v1, v2, v3)

```

```

pchit_map(data = df[, 2:4],
          well = df$wells,
          threshold = 1.5)

```

---

pc_grid	<i>Plots multiple platemaps as a heatmap of the first principal component.</i>
---------	--

---

**Description**

Converts multivariate data and well labels into a heatmap of the first principal component in the form of a grid of platemaps.

**Usage**

```
pc_grid(data, well, plate_id, ncols = 2, plate = 96, ...)
```

**Arguments**

data	Numerical values be transformed, scaled and plotted as a colour
well	Vector of well identifiers e.g "A01"
plate_id	Vector of plate labels or identifiers e.g "plate_1"
ncols	Number of columns to plot multiple platemaps
plate	Number of wells in complete plate (96, 384 or 1536)
...	additional arguments to be passed to <code>z_grid</code>

**Value**

ggplot plot

**Examples**

```
df01 <- data.frame(
  well = num_to_well(1:96),
  plate = 1,
  vals1 = rnorm(1:96),
  vals2 = rnorm(1:96))

df02 <- data.frame(
  well = num_to_well(1:96),
  plate = 2,
  vals1 = rnorm(1:96),
  vals2 = rnorm(1:96))

df <- rbind(df01, df02)

pc_grid(data = df[, 3:4],
  well = df$well,
  plate_id = df$plate,
  plate = 96)
```

---

pc_map	<i>Principal component heatmap in a plate layout</i>
--------	--

---

**Description**

Takes the values and well identifiers, calculates the first principal component, scales and plots the component as a heatmap in the form of a 96 or 384-well plate. A way to quickly show variation of multi-parametric data within a plate.

**Usage**

```
pc_map(data, well, plate = 96, ...)
```

**Arguments**

data	Vector of numerical data to calculate the first principal component
well	Vector of well identifiers e.g "A01"
plate	Number of wells in complete plate (96, 384 or 1536)
...	additional parameters to <code>platetools::z_map</code>

**Value**

gplot plot

**Examples**

```
df <- data.frame(
  well = num_to_well(1:96),
  vals1 = rnorm(1:96),
  vals2 = rnorm(1:96))

pc_map(data = df[, 2:3],
  well = df$well,
  plate = 96)
```

---

plate_effect	<i>Two way-median smooth on a plate map</i>
--------------	---

---

**Description**

Given a platemap produced by `plate_map`, this will perform a two way median smooth, and return the results of `medpolish`. Useful for row and column effects, as well as the raw residuals.

**Usage**

```
plate_effect(platemap, plate)
```

**Arguments**

platemap	platemap produced by plate_map
plate,	integer either 96 or 384

---

plate_map	<i>creates dataframe of row,column,data from wellID and data</i>
-----------	--

---

**Description**

internal function

**Usage**

```
plate_map(data, well)
```

**Arguments**

data	numeric data to be used as colour scale
well	alpha-numeric well IDs, e.g 'A01'

**Value**

dataframe

---

plate_map_grid	<i>creates dataframe of row, column, plate_id from data regarding wellIDs</i>
----------------	---

---

**Description**

internal function

**Usage**

```
plate_map_grid(data, well, plate_id)
```

**Arguments**

data	numerical data to be used as colour scale
well	alpha-numeric wellIDs, e.g 'A01'
plate_id	plate identifiers e.g 'plate_1'

**Value**

dataframe



---

plate\_map\_grid\_scale *creates dataframe of row, column, plate\_id from data regarding wellIDs*

---

**Description**

internal function

**Usage**

```
plate_map_grid_scale(data, well, plate_id, each)
```

**Arguments**

data	numerical data to be used as colour scale
well	alpha-numeric wellIDs, e.g 'A01'
plate_id	plate identifiers e.g 'plate_1'
each	boolean, if true scales each plate individually, if false will scale the pooled values of data

**Value**

dataframe

---

plate\_map\_multiple *row, column for multiple features*

---

**Description**

Generates a dataframe for multiple features, given a wellID column and multiple features

**Usage**

```
plate_map_multiple(data, well)
```

**Arguments**

data	vector or dataframe of numeric data
well	vector of alphanumeric well IDs e.g 'A01'

---

plate_map_scale	<i>creates dataframe of row, column, and scaled data from well IDs</i>
-----------------	--

---

**Description**

internal function

**Usage**

```
plate_map_scale(data, well)
```

**Arguments**

data	numeric data to be used as colour scale
well	alpha-numeric well IDs, e.g 'A01'

**Value**

dataframe

---

plate_matrix	<i>plate layout matrix from well IDs</i>
--------------	--

---

**Description**

Given a dataframe of alpha-numeric well IDs e.g ("A01"), and values, this function will produce a matrix in the form of a plate layout.

**Usage**

```
plate_matrix(data, well, plate = 96)
```

**Arguments**

data	vector of data to be placed in matrix
well	vector of alphanumeric well IDs. e.g ("A01")
plate	number of wells in plate (96 or 384)

**Value**

matrix

**Examples**

```

a <- 1:96
wells <- num_to_well(1:96)
plate_matrix(data = a, well = wells)

x <- rnorm(384)
wells <- num_to_well(1:384, plate = 384)
plate_matrix(data = x, well = wells, plate = 384)

```

plt1536

*ggplot plate object***Description**

internal function

**Usage**

```
plt1536(platemap, size = 3.5, shape = 22)
```

**Arguments**

platemap	platemap dataframe produced by <code>plate_map</code>
size	int, size parameter for <code>ggplot2::geom_point</code>
shape	int, shape parameter for <code>ggplot2::geom_point</code>

**Value**

ggplot object

plt384

*ggplot plate object***Description**

internal function

**Usage**

```
plt384(platemap, size = 5, shape = 22)
```

**Arguments**

platemap	platemap dataframe produced by <code>plate_map</code>
size	int, size parameter for <code>ggplot2::geom_point</code>
shape	int, shape parameter for <code>ggplot2::geom_point</code>

**Value**

ggplot object

---

plt96	<i>ggplot plate object</i>
-------	----------------------------

---

**Description**

internal function

**Usage**

plt96(platemap, size = 10, shape = 21)

**Arguments**

platemap	platemap dataframe produced by plate_map
size	int, size parameter for ggplot2::geom_point
shape	int, shape parameter for ggplot2::geom_point

**Value**

ggplot object

---

raw_grid	<i>Plots multiple platemaps with heatmap of raw values</i>
----------	--

---

**Description**

Converts numerical values, well labels, and plate labels into multiple plate heatmaps

**Usage**

raw\_grid(data, well, plate\_id, ncols = 2, plate = 96, ...)

**Arguments**

data	Numerical values to be plotted
well	Vector of well identifiers e.g "A01"
plate_id	Vector of plate identifiers e.g "Plate_1"
ncols	Number of columns to display multiple heatmaps
plate	Number of wells in complete plate (96, 384 or 1536)
...	additional parameters to plot wrappers

**Value**

ggplot plot

**Examples**

```
df01 <- data.frame(well = num_to_well(1:96),
  vals = rnorm(96),
  plate = 1)

df02 <- data.frame(well = num_to_well(1:96),
  vals = rnorm(96),
  plate = 2)

df <- rbind(df01, df02)

raw_grid(data = df$vals,
  well = df$well,
  plate_id = df$plate,
  plate = 96)
```

---

raw\_map

*Plots a platemap with heatmap of raw values*

---

**Description**

Converts numerical values and well labels into multiple plate heatmaps

**Usage**

```
raw_map(data, well, plate = 96, ...)
```

**Arguments**

data	Numerical values to be plotted
well	Vector of well identifiers e.g "A01"
plate	Number of wells in complete plate (96, 384 or 1536)
...	additional parameters to plot wrappers

**Value**

ggplot plot

**Examples**

```
df <- data.frame(vals = rnorm(1:384),
  well = num_to_well(1:384, plate = 384))

raw_map(data = df$vals,
  well = df$well,
  plate = 384)
```

---

readmap_data	<i>example data in a plate map form</i>
--------------	---

---

**Description**

example data in a plate map form

**Usage**

```
readmap_data
```

**Format**

96 integers structured in a the form of a 96-well plate

**Source**

none

---

read_map	<i>Annotates dataframe with metadata in a platemap matrix</i>
----------	---

---

**Description**

Annotates a dataframe contained well identifiers with metadata in the form of a platemap matrix, matching the existing well-labels to the well position in the platemap

**Usage**

```
read_map(data, map, verbose = TRUE, new_col_name = "header")
```

**Arguments**

data	existing dataframe, with wellIDs under the column name of 'well'
map	Matrix of metadata to be added to the dataframe, N.B NO MISSING WELLS!
verbose	Boolean, if TRUE will add row and column numbers to dataframe
new_col_name	What to call the added metadata

**Value**

dataframe with new column named after 'new\_col\_name'

---

rotate_plate	<i>rotates matrix by 180 degrees</i>
--------------	--------------------------------------

---

**Description**

If someone (no names) puts in a plate upside down, this function will rotate a plate matrix produced by `plate_matrix` to be the correct way up. I.e if A01 is in the bottom right hand corner rather than the top left.

**Usage**

```
rotate_plate(m)
```

**Arguments**

m	matrix
---	--------

**Value**

matrix

---

set_block	<i>Set values in rectangular areas of a plate</i>
-----------	---

---

**Description**

Updates a table representing a multiwell plate, by setting a given value for all wells in a block or a list of blocks defined by the well coordinates of their upper-left and bottom-right corners.

**Usage**

```
set_block(plate, block, what, value)
```

**Arguments**

plate	A table representing a multiwell plate, with one column named "well" representing the well identifiers.
block	Coordinates of a rectangular block (such as "A01~B02"), or a vector of coordinates.
what	A column name in the table.
value	The value to set.

**Details**

This function is tested with a table in `data.frame` format, but it should also work with the `DataFrame`, `data.table` and `tibble` formats.

**Value**

Returns the 'plate' table, where the values for the wells indicated in the blocks have been updated.

**Author(s)**

Charles Plessy

**See Also**

[num\\_to\\_well](#)

**Examples**

```
p <- data.frame(well = num_to_well(1:96))
head(p)

p <- set_block(p, c("A01~B02", "A05~D05"), "dNTP", 0.25)
p <- set_block(p, "A03", "dNTP", 0.50)
head(p)

# Be careful with the column names
p <- set_block(p, "A01~H12", "Mg2+", 3.0)
head(p)

## Not run:
# Chained updates with magrittr
p %<>%
  setBlock("A01~C04", "dNTP", 0.5) %>%
  setBlock("A01~C04", "Mg", 3.0)

## End(Not run)
```

---

well\_to\_num

*Converts well labels to numbers*

---

**Description**

Converts alpha-numeric well labels to numbers corresponding to positions within a microtitre plate. Either 96 or 384 well plate, in column-wise order or in a column snaking pattern.

**Usage**

```
well_to_num(wells, style = "normal", plate = 96)
```



**Arguments**

wells	Vector of well identifiers e.g "A01"
style	Either normal, starting at the left hand column at each row or in a snaking fashion. ('normal' or 'snake')
plate	Number of wells in the complete plate (96 or 384)

**Value**

Vector of numbers

**Examples**

```
well_to_num("A01")

well_to_num("P12", plate = 384)

well_to_num("P12", plate = 384, style = "snake")

wells <- c("A01", "A02", "A03")
well_to_num(wells)
```

---

z\_grid

*Plots multiple platemaps with heatmap of scaled values*


---

**Description**

Converts numerical values, well labels, and plate labels into multiple plate heatmaps

**Usage**

```
z_grid(data, well, plate_id, ncols = 2, plate = 96, each = FALSE,
       scale_each = FALSE, ...)
```

**Arguments**

data	Numerical values to be plotted
well	Vector of well identifiers e.g "A01"
plate_id	Vector of plate identifiers e.g "Plate_1"
ncols	Number of columns to display multiple heatmaps
plate	Number of wells in complete plate (96, 384 or 1569)
each	boolean, allowed for backwards compatibility, scale_each is now the preferred argument name
scale_each	boolean, if true scales each plate individually, if false will scale the pooled values of data
...	additional parameters to plot wrappers

**Value**

ggplot plot

**Examples**

```
df01 <- data.frame(well = num_to_well(1:96),  
  vals = rnorm(96),  
  plate = 1)
```

```
df02 <- data.frame(well = num_to_well(1:96),  
  vals = rnorm(96),  
  plate = 2)
```

```
df <- rbind(df01, df02)
```

```
z_grid(data = df$vals,  
  well = df$well,  
  plate_id = df$plate,  
  plate = 96)
```

---

z\_map

*Plots a platemap with heatmap of scaled values*

---

**Description**

Converts numerical values and well labels into multiple plate heatmaps

**Usage**

```
z_map(data, well, plate = 96, ...)
```

**Arguments**

data	Numerical values to be plotted
well	Vector of well identifiers e.g "A01"
plate	Number of wells in complete plate (96, 384 or 1536)
...	additional parameters to plot wrappers

**Value**

ggplot plot

**Examples**

```
df <- data.frame(vals = rnorm(1:384),  
  well = num_to_well(1:384, plate = 384))
```

```
z_map(data = df$vals,  
  well = df$well,  
  plate = 384)
```

# Index

## \*Topic **datasets**

readmap\_data, 22

b\_grid, 3

b\_map, 4

b\_score, 5

bhit\_map, 2

check\_plate\_input, 6

data.frame, 24

data.table, 24

DataFrame, 24

dist\_map, 6

fill\_plate, 7

hit\_grid, 7

hit\_map, 8

is\_1536, 9

list\_to\_dataframe, 10

med\_smooth, 10

missing\_wells, 11

num\_to\_well, 11, 24

pc\_grid, 14

pc\_map, 15

pchit\_grid, 12

pchit\_map, 13

plate\_effect, 15

plate\_map, 16

plate\_map\_grid, 16

plate\_map\_grid\_scale, 17

plate\_map\_multiple, 17

plate\_map\_scale, 18

plate\_matrix, 18

plt1536, 19

plt384, 19

plt96, 20

raw\_grid, 20

raw\_map, 21

read\_map, 22

readmap\_data, 22

rotate\_plate, 23

set\_block, 23

tibble, 24

well\_to\_num, 24

z\_grid, 25

z\_map, 26