Package

March 12, 2019

Type Package

Title What The Package Does (one line, title case required)

Version 0.1.10

Description The description of a package is usually long,
spanning multiple lines. The second and subsequent lines
should be indented, usually with four spaces.

Depends R (>= 3.5.0)

Imports spatstat, tools, tiff, h5, raster, igraph, plyr, dplyr, readr

Suggests testthat, knitr, rmarkdown, roxygen2, devtools

License What license is it under?

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

VignetteBuilder knitr

R topics documented:

`.assign_module` *Pixel module Function assign a module to each dead pixel*

Description

Pixel module Function assign a module to each dead pixel

Usage

```
.assign_module(layout)
```

Arguments

`layout`	Layout object

Value

dead_modules

`.classify_clump` *Clasifies a clump*

Description

Clasifies a clump

Usage

```
.classify_clump(layout, x, y)
```

Arguments

`x`	vector containing the x coordinates of a clump
`y`	vector containing the y coordinates of a clump

Value

the class of a clump (1 - singleton, 2 - double, 3 - triplet, 4 - larger cluster, unless it actually has the shape of a line, 5 (6): vertical line where closest edge is the upper (lower) one, 7 (8): horizontal line where closest edge is the right (left) one)

`.clump_module` *Identifying modules for clumps*

Description

Identifying modules for clumps

Usage

```
.clump_module(layout, rrc)
```

Arguments

`layout`	Layout object
`rrc`	raster clumps objects

`.dist_edge` *Function returns distance to both edges. Given xy coo of pixel and matrices with positions of edges in both directions.*

Description

Function returns distance to both edges. Given xy coo of pixel and matrices with positions of edges in both directions.

Usage

```
.dist_edge(xy, module_edges)
```

Arguments

`xy`	?
`module_edges`	?

Value

tmp ?

`.extract_layout_parameter` *Checks whether a layout parameter is in the file string*

Description

Checks whether a layout parameter is in the file string

Usage

```
.extract_layout_parameter(file_string, parameter)
```

Arguments

`file_string`	String of a file context
`parameter`	Layout parameter

Value

parameter value

`.extract_number` *Internal function to convert string values to numbers*

Description

Internal function to convert string values to numbers

Usage

```
.extract_number(s)
```

Arguments

`s`	String expression?

Value

Numeric value

`.getmode` *Returns the mode of a set of data*

Description

Returns the mode of a set of data

Usage

```
.getmode(v)
```

Arguments

`v`	set of data

Value

uniqv the value of the mode

.get_clump_event_ppp *Creates ppp for damaged layout events*

Description

Creates ppp for damaged layout events

Usage

```
.get_clump_event_ppp(layout, incl_event_list = NA, height = NULL,
  width = NULL)
```

Arguments

`layout`	Layout object
`incl_event_list`	a list of events to be included

.get_clump_pixel_ppp *Creates ppp for damaged layout pixels*

Description

Creates ppp for damaged layout pixels

Usage

```
.get_clump_pixel_ppp(layout, incl_event_list = NA)
```

Arguments

`layout`	Layout object
`incl_event_list`	a list of events to be included

`.get_layout_ppps` *Generate layout ppps*

Description

Generate layout ppps

Usage

```
.get_layout_ppps(layout)
```

Arguments

`layout`	Layout object

Value

a list of ppps for edges and gaps

`.get_ppp_dead_module` *Generates ppp for the dead pixels for a selected module*

Description

Generates ppp for the dead pixels for a selected module

Usage

```
.get_ppp_dead_module(layout, row, col)
```

Arguments

`layout`	Layout object
`row`	module row number
`col`	module column number

Value

ppp of dead pixels

`.mask_to_events` *Converts mask (dead pixels) to events*

Description

Converts mask (dead pixels) to events

Usage

```
.mask_to_events(layout, dead_pix_mask, row = NA, col = NA)
```

Arguments

`layout`	Layout object
`dead_pix_mask`	Dead pixels mask

Value

list of pixels and events

`.tr` *Utils module Calculates the trace value of a square matrix*

Description

Utils module Calculates the trace value of a square matrix

Usage

```
.tr(m)
```

Arguments

`m`	A square matrix

Value

tr The trace value

`.xyc_pixels2events` *MODIFYING CLUSTERS TO EVENTS (CONSISTING OF 1 PIXEL REPRESENTING THE CLUSTER) Make into a point pattern of just events rather than pixels. Using xyc_ply object. Collapse in one point using centres for clusters, but end points for lines, type dependend: type 5 (closest to upper edge): ymin type 6 (closest to lower edge): ymax type 7 (closest to right edge): xmin type 8 (closest to left edge): xmax This is inspired by Perkin Elmer Layout and be replaced by other choices if desired.*

Description

MODIFYING CLUSTERS TO EVENTS (CONSISTING OF 1 PIXEL REPRESENTING THE CLUSTER) Make into a point pattern of just events rather than pixels. Using xyc_ply object. Collapse in one point using centres for clusters, but end points for lines, type dependend: type 5 (closest to upper edge): ymin type 6 (closest to lower edge): ymax type 7 (closest to right edge): xmin type 8 (closest to left edge): xmax This is inspired by Perkin Elmer Layout and be replaced by other choices if desired.

Usage

```
.xyc_pixels2events(xyc_ply)
```

Arguments

`xyc_ply`	clums data frame

Value

events

`.xyc_ply_func` *Something something dark side*

Description

Something something dark side

Usage

```
.xyc_ply_func(layout, xyc_pixel_df)
```

Arguments

`layout`	Layout object

Value

data frame

`check_layout_avail` *Checks whether specified layout is available*

Description

Checks whether specified layout is available

Usage

```
check_layout_avail(layout_name)
```

Arguments

`layout_name`	The name of the layout

Value

True or False

`create_module` *Checks whether layout is available, if so, creates a Layout object*

Description

Checks whether layout is available, if so, creates a Layout object

Usage

```
create_module(layout_name)
```

Arguments

`layout_name`	The name of the layout

Value

Layout object

`create_ppp_edges_col` *TODO: Better description of the function This is the ppp_edges_col function*

Description

TODO: Better description of the function This is the ppp_edges_col function

Usage

```
create_ppp_edges_col(layout)
```

Arguments

`layout`	Layout object

Value

Point pattern dataset

`create_ppp_edges_row` *TODO: Better description of the function This is the create_ppp_edges_row function*

Description

TODO: Better description of the function This is the create_ppp_edges_row function

Usage

```
create_ppp_edges_row(layout)
```

Arguments

`layout`	Layout object

Value

Point pattern dataset

`create_ppp_gaps_col` *TODO: Better description of the function This is a function*

Description

TODO: Better description of the function This is a function

Usage

```
create_ppp_gaps_col(layout)
```

Arguments

`layout`	Layout object

Value

Point pattern dataset

`create_ppp_gaps_row` *TODO: Better description of the function This is a function*

Description

TODO: Better description of the function This is a function

Usage

```
create_ppp_gaps_row(layout)
```

Arguments

`layout`	Layout object

Value

Point pattern dataset

`dead_pix_coords` *Extracts a table of dead pixel coordinates from a pixel matrix*

Description

Extracts a table of dead pixel coordinates from a pixel matrix

Usage

```
dead_pix_coords(pix_matrix)
```

Arguments

`pix_matrix`	pixel matrix with dead pixels flagged with 1

Value

Table containing dead pixel coordinates

`Dead_Stats` *Analysis module A S3 class to represent dead pixels statistics summary*

Description

Analysis module A S3 class to represent dead pixels statistics summary

Usage

```
Dead_Stats(dead_n = NA, module_n = NA, module_count_arr = NA,
  module_count = NA, avg_dead_mod = NA, Chisq_s = NA,
  Chisq_df = NA, Chisq_p = NA)
```

Arguments

`dead_n`	Total number of damaged pixels:
`module_n`	Total number of modules
`module_count_arr`	Count of dead pixels in each quadrat
`module_count`	Count of dead pixels in each quadrat
`avg_dead_mod`	Average number of damaged pixels per module
`Chisq_s`	The Chi-Squared test statistic value
`Chisq_df`	Chi-Squared degrees of freedom
`Chisq_p`	Chi-Squared p-value

Value

Dead_Stats object

dead_stats_summary *Count number of damaged pixels overall and in different modules*

Description

Count number of damaged pixels overall and in different modules

Usage

```
dead_stats_summary(layout)
```

Arguments

layout	Layout object

Value

A string with damaged pixels overall statitics

Default_Layout *Layout module A S3 class to represent a detector layout.*

Description

Layout module A S3 class to represent a detector layout.

Usage

```
Default_Layout(name = "Default", date = NA, detector_width = NA,
  detector_height = NA, module_col_n = NA, module_row_n = NA,
  module_col_sizes = NA, module_row_sizes = NA, gap_col_sizes = NA,
  gap_row_sizes = NA, module_edges_col = NA, module_edges_row = NA,
  detector_inconsistency = NA, pix_matrix = NA, pix_dead = NA,
  dead_stats = NA, pix_dead_modules = NA, clumps = NA)
```

Arguments

name	detector's name
date	date
detector_width	detector's width
detector_height	detector's height
module_col_n	number of columns in the grid of modules
module_row_n	number of rows in the grid of modules
module_col_sizes	vector with widths of the modules
module_row_sizes	vector with heights of the modules

`gap_col_sizes` vector with widths of the gaps

`gap_row_sizes` vector with heights of the gaps

`module_edges_col`
!

`module_edges_row`
!

`detector_inconsistency`
counts inconsistencies found in parameters entered

`pix_matrix` pixel matrix

`pix_dead` dead pixels coordinates

`dead_stats` dead pixel statistics

`pix_dead_modules`
assigned module for each dead pixel

`clumps` clumps data (xyc_df data frame with pixels and their clump ID's, xyc_events data frame with clusters (clumps) and their clump ID's and centre coordinates)

Value

Layout object

`derive_layout` *Deriving additional layout elements*

Description

Conditions additional elements of Layout object that are frequently used later They are calculated from parameters defined in examples Matrices that contains xy coordiantes of edges of modules By definition, edges are part of modules (not part of gaps) i.e. for each module two pairs: first/last col and first/last row.

Usage

```
derive_layout(layout)
```

Arguments

`layout` Layout object

Value

Layout object

`dist_closest_edge` *A function to calclutate closest distance to an edge*

Description

A function to calclutate closest distance to an edge

Usage

```
dist_closest_edge(x, size)
```

Arguments

`x`	something
`size`	something else

Value

what does this mean?

`dist_corner` *A function to calculate pixel distances from corners*

Description

A function to calculate pixel distances from corners

Usage

```
dist_corner(layout)
```

Arguments

`layout`	Layout object

Value

Matrix containing parallel maxima from the centre for each pixel

`dist_edge_col` *A function to calculate pixel distances from edges by column*

Description

A function to calculate pixel distances from edges by column

Usage

```
dist_edge_col(layout)
```

Arguments

`layout`	Layout object

Value

dist ?

`dist_edge_min` *A function to calculate pixel distances from edges*

Description

A function to calculate pixel distances from edges

Usage

```
dist_edge_min(layout)
```

Arguments

`layout`	Layout object

Value

dist ?

`dist_edge_row` *A function to calculate pixel distances from edges by row*

Description

A function to calculate pixel distances from edges by row

Usage

```
dist_edge_row(layout)
```

Arguments

`layout`	Layout object

Value

dist ?

`dist_vec` *Estimates the distance between vectors v and w*

Description

Estimates the distance between vectors v and w

Usage

```
dist_vec(v, w)
```

Arguments

`v`	vector
`w`	vector

Value

distance between vectors v and w

`Excalibur_Layout` *A S3 class to represent the Excalibur detector layout.*

Description

A S3 class to represent the Excalibur detector layout.

Usage

```
Excalibur_Layout()
```

Value

Excalibur layout object

`find_clumps` *Locates and clusifies clumps of a damaged layout*

Description

Locates and clusifies clumps of a damaged layout

Usage

```
find_clumps(layout, row = NA, col = NA)
```

Arguments

`layout`	Layout object
`row`	Module row number
`col`	Module column number

`get_dead_pix_mask` *Creates a mask matrix of dead pixels*

Description

Creates a mask matrix of dead pixels

Usage

```
get_dead_pix_mask(layout)
```

Arguments

`layout`	Layout object

Value

dead pixel mask

`get_dead_stats` *Count number of damaged pixels overall and in different modules*

Description

Count number of damaged pixels overall and in different modules

Usage

```
get_dead_stats(layout)
```

Arguments

`layout`	Layout object

Value

Dead_Stats object

`get_events_mask` *Generates events matrix (a matrix with pixels as 0 and events as 1)*

Description

Generates events matrix (a matrix with pixels as 0 and events as 1)

Usage

```
get_events_mask(layout)
```

Arguments

`layout`	Layout object

Value

events mask

`get_ppp_dead` *Generates ppp for the dead pixels*

Description

Generates ppp for the dead pixels

Usage

```
get_ppp_dead(layout)
```

Arguments

`layout`	Layout object

Value

ppp of dead pixels

`glm_pixel_ctr_eucl` *Fits pixel distance from the centre to*

Description

Fits pixel distance from the centre to

Usage

```
glm_pixel_ctr_eucl(layout)
```

Arguments

`layout`	Layout object

Value

Fitted model

`glm_pixel_ctr_linf` *Fits pixel parallel maxima from the centre*

Description

Fits pixel parallel maxima from the centre

Usage

```
glm_pixel_ctr_linf(layout)
```

Arguments

`layout`	Layout object

Value

Fitted model

`glm_pixel_dist_edge_col` *Fits pixel istances from the module edges by column*

Description

Fits pixel istances from the module edges by column

Usage

```
glm_pixel_dist_edge_col(layout)
```

Arguments

`layout`	Layout object

Value

Fitted model

glm_pixel_dist_edge_row *Fits pixel istances from the module edges by row*

Description

Fits pixel istances from the module edges by row

Usage

```
glm_pixel_dist_edge_row(layout)
```

Arguments

`layout`	Layout object

Value

Fitted model

inconsist_dead_layout *Counts damaged pixel locations (dead_data) outside detector (layout) and in gaps between modules and give warnings*

Description

Counts damaged pixel locations (dead_data) outside detector (layout) and in gaps between modules and give warnings

Usage

```
inconsist_dead_layout(dead_data, layout)
```

Arguments

`dead_data`	Dead pixel locations
`layout`	Layout object

Value

Inconsistency message

`ini_graphics` *Starts the graphics device driver for producing graphics with respect to a chosen format*

Description

Starts the graphics device driver for producing graphics with respect to a chosen format

Usage

```
ini_graphics(file_path)
```

Arguments

`file_path`	Output path with an extension

`layout_consist_check` *Basic checks if parameters entered (slightly redundant on purpose) add up*

Description

Basic checks if parameters entered (slightly redundant on purpose) add up

Usage

```
layout_consist_check(layout = NA)
```

Arguments

`layout`	Layout object

`layout_edges` *Defines the coordinates of layout's edges using module and gap sizes*

Description

Function is in 1d context to be applied to rows and cols separately. Edges are inside the modules (first/last row/col of module).

Usage

```
layout_edges(m, g)
```

Arguments

`m`	vector of module sizes
`g`	vectors of gap sizes

Value

Matrix with the information about the edges

`layout_summary` *Returns a string with the layout summary*

Description

Returns a string with the layout summary

Usage

```
layout_summary(layout)
```

Arguments

`layout`	Layout object

Value

String with the layout summary

`load_pix_matrix` *A function to load pixel data*

Description

A function to load pixel data

Usage

```
load_pix_matrix(layout, file_path)
```

Arguments

`layout`	The name of the layout to be used
`file_path`	Path(s) to the file(s) containing dead pixel information

Value

Layout object

`matrix_from_hdf` *Reads in hdf file(s) and returns a pixel matrix*

Description

Reads in hdf file(s) and returns a pixel matrix

Usage

```
matrix_from_hdf(layout, file_path)
```

Arguments

`layout`	Layout object
`file_path`	A list of paths to hdf files. Must be in the correct order.

Value

Data of a combined dataset from hdf files

`matrix_from_tiff` *I/O module Reads in tiff file and returns a pixel matrix*

Description

I/O module Reads in tiff file and returns a pixel matrix

Usage

```
matrix_from_tiff(layout, file_path)
```

Arguments

`layout`	Layout object
`file_path`	Path to the tiff file

Value

Pixel matrix with dead pixels flagged with 1

matrix_from_xml *Reads in xml file and returns a pixel matrix*

Description

Reads in xml file and returns a pixel matrix

Usage

```
matrix_from_xml(layout, file_path)
```

Arguments

`layout`	Layout object
`file_path`	Path to the xml file

Value

Data from an xml file

norm_vec *Estimates the norm of a vector*

Description

Estimates the norm of a vector

Usage

```
norm_vec(v)
```

Arguments

`v`	vector

Value

norm of the vector v

`orientnnPPP` *Get orient nn PP*

Description

Get orient nn PP

Usage

```
orientnnPPP(PPPdata)
```

Arguments

`PPPdata`	describe

Value

describe

`orient_dist_vec` *Calculates distance and orientation of the oriented vector between two points in order of the second pointing to first (reflecting nearest neighbour (nn) framework) v, w point coordinates indicating vectors wrt to the origin. Values: distance and orientation (in [0,360) degrees) of w pointing towards v.*

Description

Calculates distance and orientation of the oriented vector between two points in order of the second pointing to first (reflecting nearest neighbour (nn) framework) v, w point coordinates indicating vectors wrt to the origin. Values: distance and orientation (in [0,360) degrees) of w pointing towards v.

Usage

```
orient_dist_vec(v, w)
```

Arguments

`v`	vector
`w`	vector

Value

distance and orientation of the oriented vector between two points

`perform_glm` *Performs model fitting on the specified symbolic expression*

Description

Performs model fitting on the specified symbolic expression

Usage

```
perform_glm(symb_expr, family = binomial(link = logit))
```

Arguments

`symb_expr`	symbolic description of the linear predictor
`family`	a description of the error distribution

Value

Fitted model

glm_git fitted model

`PerkinElmerCropped1600_Layout` *A S3 class to represent the PerkinElmerCropped1600 detector layout.*

Description

A S3 class to represent the PerkinElmerCropped1600 detector layout.

Usage

```
PerkinElmerCropped1600_Layout()
```

Value

PerkinElmerCropped1600 layout object

`PerkinElmerFull_Layout` *A S3 class to represent the PerkinElmerFull detector layout.*

Description

A S3 class to represent the PerkinElmerFull detector layout.

Usage

```
PerkinElmerFull_Layout()
```

Value

PerkinElmerFul layout object

`PerkinElmerRefurbished_Layout` *A S3 class to represent the PerkinElmerRefurbished detector layout.*

Description

A S3 class to represent the PerkinElmerRefurbished detector layout.

Usage

```
PerkinElmerRefurbished_Layout()
```

Value

PerkinElmerRefurbished layout object

`Pilatus_Layout` *A S3 class to represent the PerkinElmerRefurbished detector layout.*

Description

A S3 class to represent the PerkinElmerRefurbished detector layout.

Usage

```
Pilatus_Layout()
```

Value

Pilatus layout object

`pixel_dist_ctr_eucl` *A function to calculate euclidean distance from the centre*

Description

A function to calculate euclidean distance from the centre

Usage

```
pixel_dist_ctr_eucl(layout)
```

Arguments

`layout`	Layout object

Value

Matrix containing euclidean distances from the centre for each pixel

`pixel_dist_ctr_linf` *A function to calculate parallel maxima from the centre*

Description

A function to calculate parallel maxima from the centre

Usage

```
pixel_dist_ctr_linf(layout)
```

Arguments

`layout`	Layout object

Value

Matrix containing parallel maxima from the centre for each pixel

plot_angles *ANGLES using nnorient() from spatstat package*

Description

ANGLES using nnorient() from spatstat package

Usage

```
plot_angles(ppp_obj, caption, file_path = NA)
```

Arguments

ppp_obj	ppp object
caption	caption of the figure
file_path	file path

plot_arrows *A function to plot NN oriented arrrows*

Description

A function to plot NN oriented arrrows

Usage

```
plot_arrows(ppp_obj, caption, file_path = NA)
```

Arguments

ppp_obj	ppp object
caption	caption of the figure
file_path	file path

`plot_counts` *A function to plot NN oriented arrrows*

Description

A function to plot NN oriented arrrows

Usage

```
plot_counts(module_count_arr, caption, file_path = NA)
```

Arguments

`module_count_arr`	Counts per array
`caption`	caption of the figure
`file_path`	file path

`plot_density` *Plots module Plots ppp object's density*

Description

Plots module Plots ppp object's density

Usage

```
plot_density(ppp_obj, caption, file_path = NA, adjust = 0.25)
```

Arguments

`ppp_obj`	ppp object
`caption`	caption of the figure
`file_path`	file path
`adjust`	Kernel density bandwidth

`plot_events` *Plots damaged layout events*

Description

Plots damaged layout events

Usage

```
plot_events(layout, file_path = NA, caption = TRUE,
  incl_event_list = NA, plot_edges_gaps = TRUE)
```

Arguments

`layout` Layout object

`file_path` Output file path

`caption` Flag to turn on/off figure caption

`incl_event_list`
a list of events to be included

`plot_events_angles` *Plots angles graph of events*

Description

Plots angles graph of events

Usage

```
plot_events_angles(layout, file_path = NA, row = NA, col = NA,
  caption = TRUE, incl_event_list = NA)
```

Arguments

`layout` Layout object

`file_path` Output file path

`row` Module row number

`col` Module column number

`caption` Flag to turn on/off figure caption

`incl_event_list`
a list of events to be included

plot_events_arrows *Plots arrows graph of events*

Description

Plots arrows graph of events

Usage

```
plot_events_arrows(layout, file_path = NA, row = NA, col = NA,
  caption = TRUE, incl_event_list = NA)
```

Arguments

layout	Layout object
file_path	Output file path
row	Module row number
col	Module column number
caption	Flag to turn on/off figure caption
incl_event_list	a list of events to be included

plot_events_count *A function to plot layout with counts per module*

Description

A function to plot layout with counts per module

Usage

```
plot_events_count(layout, file_path = NA, row = NA, col = NA,
  caption = TRUE, incl_event_list = NA)
```

Arguments

layout	Layout object
file_path	Output file path
row	Module row number
col	Module column number
caption	Flag to turn on/off figure caption

`plot_events_density` *Plots density graph of events*

Description

Plots density graph of events

Usage

```
plot_events_density(layout, file_path = NA, adjust = 0.25, row = NA,
  col = NA, caption = TRUE, incl_event_list = NA)
```

Arguments

`layout`	Layout object
`file_path`	Output file path
`adjust`	Kernel density bandwidth
`row`	Module row number
`col`	Module column number
`caption`	Flag to turn on/off figure caption
`incl_event_list`	a list of events to be included

`plot_events_kfg` *Plots K, F, G functions*

Description

Plots K, F, G functions

Usage

```
plot_events_kfg(layout, func, file_path = NA, row = NA, col = NA,
  caption = TRUE, incl_event_list = NA)
```

Arguments

`layout`	Layout object
`func`	Function name
`file_path`	Output file path
`row`	Module row number
`col`	Module column number
`caption`	Flag to turn on/off figure caption
`incl_event_list`	a list of events to be included

`plot_kfg` *Plots K, F, G functions*

Description

Plots K, F, G functions

Usage

```
plot_kfg(ppp_obj, func, file_path = NA, caption = TRUE)
```

Arguments

`ppp_obj`	ppp object
`func`	Function name
`file_path`	Output file path
`caption`	Flag to turn on/off figure caption

`plot_layout` *Plotting layout*

Description

Plotting layout

Usage

```
plot_layout(layout, file_path = NA, caption = TRUE)
```

Arguments

`layout`	Layout object
`file_path`	Output file path
`caption`	Flag to turn on/off figure caption

`plot_layout_angles` *ANGLES using nnorient() from spatstat package*

Description

ANGLES using nnorient() from spatstat package

Usage

```
plot_layout_angles(layout, file_path = NA, row = NA, col = NA,
  caption = TRUE)
```

Arguments

`layout`	Layout object
`file_path`	Output file path
`row`	Module row number
`col`	Module column number
`caption`	Flag to turn on/off figure caption

`plot_layout_arrows` *A function to plot NN oriented arrrows*

Description

A function to plot NN oriented arrrows

Usage

```
plot_layout_arrows(layout, file_path = NA, row = NA, col = NA,
  caption = TRUE)
```

Arguments

`layout`	Layout object
`file_path`	Output file path
`row`	Module row number
`col`	Module column number
`caption`	Flag to turn on/off figure caption

plot_layout_cnt_mod *A function to plot layout with counts per module*

Description

A function to plot layout with counts per module

Usage

```
plot_layout_cnt_mod(layout, file_path = NA, row = NA, col = NA,
  caption = TRUE)
```

Arguments

layout	Layout object
file_path	Output file path
row	Module row number
col	Module column number
caption	Flag to turn on/off figure caption

plot_layout_damaged *A function to plot layout with damaged pixels*

Description

A function to plot layout with damaged pixels

Usage

```
plot_layout_damaged(layout, file_path = NA, caption = TRUE)
```

Arguments

layout	Layout object
file_path	Output file path
caption	Flag to turn on/off figure caption

`plot_layout_density` *A function to plot layout with dead pixel densities*

Description

A function to plot layout with dead pixel densities

Usage

```
plot_layout_density(layout, file_path = NA, adjust = 0.25, row = NA,
  col = NA, caption = TRUE)
```

Arguments

`layout`	Layout object
`file_path`	Output file path
`adjust`	Kernel density bandwidth
`row`	Module row number
`col`	Module column number
`caption`	Flag to turn on/off figure caption

`plot_layout_kfg` *Plots K, F, G functions*

Description

Plots K, F, G functions

Usage

```
plot_layout_kfg(layout, func, file_path = NA, row = NA, col = NA,
  caption = TRUE)
```

Arguments

`layout`	Layout object
`func`	Function name
`file_path`	Output file path
`row`	module row number
`col`	module column number
`caption`	Flag to turn on/off figure caption

plot_layout_module_damaged *Plotting a module of a layout*

Description

Plotting a module of a layout

Usage

```
plot_layout_module_damaged(layout, col, row, file_path = NA,
  caption = TRUE)
```

Arguments

layout	Layout object
file_path	Output file path
caption	Flag to turn on/off figure caption
mod_col	Module column number
mod_row	Module row number

plot_module_events *Plots damaged layout module events*

Description

Plots damaged layout module events

Usage

```
plot_module_events(layout, col, row, file_path = NA, caption = TRUE,
  incl_event_list = NA)
```

Arguments

layout	Layout object
file_path	Output file path
caption	Flag to turn on/off figure caption
incl_event_list	a list of events to be included
mod_col	Module column number
mod_row	Module row number

plot_pixel *Plots pixel analysis*

Description

Plots pixel analysis

Usage

```
plot_pixel(data, width, height, file_path = NA)
```

Arguments

`data`	Matrix containing pixel analysis data
`width`	Plot width
`height`	Plot height
`file_path`	Output path with an extension

plot_pixels_events *Plots damaged layout pixels and events*

Description

Plots damaged layout pixels and events

Usage

```
plot_pixels_events(layout, file_path = NA, caption = TRUE,
  incl_event_list = NA)
```

Arguments

`layout`	Layout object
`file_path`	Output file path
`caption`	Flag to turn on/off figure caption
`incl_event_list`	a list of events to be included

plot_pixel_ctr_eucl *Calculates and plots pixel euclidean distance from the centre*

Description

Calculates and plots pixel euclidean distance from the centre

Usage

```
plot_pixel_ctr_eucl(layout, file_path = NA)
```

Arguments

layout	Layout object
file_path	Output file path

plot_pixel_ctr_linf *Calculates and plots pixel parallel maxima from the centre*

Description

Calculates and plots pixel parallel maxima from the centre

Usage

```
plot_pixel_ctr_linf(layout, file_path = NA)
```

Arguments

layout	Layout object
file_path	Output file path

plot_pixel_dist_corner *Calculates and plots pixel distances from corners*

Description

Calculates and plots pixel distances from corners

Usage

```
plot_pixel_dist_corner(layout, file_path = NA)
```

Arguments

layout	Layout object
file_path	Output file path

`plot_pixel_dist_edge` *Calculates and plots minimum distances from the module edges*

Description

Calculates and plots minimum distances from the module edges

Usage

```
plot_pixel_dist_edge(layout, file_path = NA)
```

Arguments

`layout`	Layout object
`file_path`	Output file path

`plot_pixel_dist_edge_col` *Calculates and plots distances from the module edges by column*

Description

Calculates and plots distances from the module edges by column

Usage

```
plot_pixel_dist_edge_col(layout, file_path = NA)
```

Arguments

`layout`	Layout object
`file_path`	Output file path

`plot_pixel_dist_edge_row` *Calculates and plots distances from the module edges by row*

Description

Calculates and plots distances from the module edges by row

Usage

```
plot_pixel_dist_edge_row(layout, file_path = NA)
```

Arguments

`layout`	Layout object
`file_path`	Output file path

`readin_layout` *Reads in a user defined layout from a file*

Description

Reads in a user defined layout from a file

Usage

```
readin_layout(file_path)
```

Arguments

`file_path`	A path to the user defined layout file

Value

Layout object

`which_module` *Module module Which module function*

Description

Module module Which module function

Usage

```
which_module(coo, me)
```

Arguments

`coo`	?
`me`	?

Value

which_module what does this mean?

`which_module_idx` *Function returns both col and row wrt layout grid. Given xy coo of pixel and matrices with positions of edges in both directions.*

Description

Function returns both col and row wrt layout grid. Given xy coo of pixel and matrices with positions of edges in both directions.

Usage

```
which_module_idx(x, y, module_edges_col, module_edges_row)
```

Arguments

`x`	?
`y`	?
`module_edges_col`	?
`module_edges_row`	?

Value

tmp ?

Index