

Package: gfplot (via r-universe)

September 16, 2026

Type Package

Title Publication-Ready Figures for Cancer Bioinformatics

Version 0.6.0

Description A compact plotting toolkit for cancer bioinformatics figures. Provides publication-ready Kaplan-Meier curves with hazard ratios, ROC and time-dependent ROC curves, multiple-ROC comparisons, risk-score distributions, PCA and UMAP projections, boxplots and correlation plots, lasso coefficient paths, enrichment plots, and immune-infiltration radar charts. Palette helpers return colour-blind-friendly colours in journal styles.

License Apache License (≥ 2)

URL <https://github.com/gflab/gfplot>, <https://gflab.github.io/gfplot/>

BugReports <https://github.com/gflab/gfplot/issues>

Encoding UTF-8

Imports cli, cowplot, ggsci, ggplot2, ggrepel, pROC, RColorBrewer, rlang, scales, survival, survminer, tibble, tidy

Suggests clinstats, digest, extrafont, fgsea, ggpubr, ggsignif, grid, glmnet, ggradar, HTSanalyzeR2, knitr, ragg, rmarkdown, Rtsne, survivalROC, systemfonts, testthat ($\geq 3.0.0$), timeROC, umap, withr

Enhances DOSE

Remotes ricardo-bion/ggradar

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.0

Config/pak/sysreqs cmake make libicu-dev libjpeg-dev libpng-dev libxml2-dev libssl-dev

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

Date/Publication 2026-09-16 05:20:13 UTC

RemoteUrl <https://github.com/gflab/gfplot>

RemoteRef master

RemoteSha d72c92e414008834897ea847f417633492b7f5f6

Contents

generate_time_event	3
get_color	3
gfplot_colors	4
gfplot_families	5
gfplot_font_setup	5
gfplot_legend	6
gfplot_render	7
gfplot_save	8
gfplot_spec	9
gfplot_theme	10
ggGSEA	11
plot_barplot	12
plot_Boxplot	13
plot_calibration	14
plot_confusion	15
plot_cor	16
plot_cumulative_incidence	17
plot_decision_curve	18
plot_embedding	19
plot_forest	20
plot_GO	22
plot_heatmap	23
plot_immune	24
plot_KMCurve	25
plot_lasso	26
plot_MulROC	27
plot_PCA	28
plot_pr_curve	29
plot_RiskScore	30
plot_ROC	31
plot_TimeROC	32
plot_tsne	33
plot_UMAP	34
plot_violin	35
plot_volcano	36
plot_waterfall	37
viewGSEA	38

Index

40

generate_time_event	<i>Build event indicators for one or more follow-up limits</i>
---------------------	--

Description

Turns a two-column clinical matrix into a matrix of event indicators, with one column per follow-up limit. Observations beyond a limit are censored at that limit.

Usage

```
generate_time_event(clinical, limits, labels = NULL)
```

Arguments

clinical	Two-column object: follow-up time and event indicator.
limits	Numeric vector of follow-up limits.
labels	Optional column names for the resulting matrix.

Value

A matrix with one column per limit.

Examples

```
clinical <- cbind(time = c(1, 5, 10), event = c(1, 0, 1))
generate_time_event(clinical, limits = c(3, 6), labels = c("year3", "year6"))
```

get_color	<i>Colour palettes used by the plotting functions</i>
-----------	---

Description

Returns a vector of journal-style colours. When palette is a vector of length greater than one it is returned unchanged, so the argument can also be used to pass colours through from a caller.

Usage

```
get_color(palette, n = 6)
```

Arguments

palette	Palette name: "nature", "jco", "lancet", "jama", "jama_classic", "house" for the lab series ramp described in gfplot_colors() , or any other value, which selects the ColorBrewer "Set1" palette.
n	Number of colours to return.

Value

A character vector of colours.

Examples

```
get_color("jama", 3)
get_color("house", 4)
get_color(c("#111111", "#222222"))
```

gfplot_colors

House colours by semantic role

Description

Returns the colours gfplot uses, addressed by role rather than by hex value. Override any role for a session with `options(gfplot.palette = list(primary = "#123456"))`, which is how the suite is retuned without editing figures.

Usage

```
gfplot_colors(role = "all")
```

Arguments

role	Colour roles to return. Use "all" for every role, or "series" for the ordered ramp that multi-series figures walk. Aliases such as "model_curve", "comparator_curve", and "reference_line" map onto the same roles.
------	---

Value

A named character vector of colours.

See Also

[get_color\(\)](#) for journal palettes and [gfplot_theme\(\)](#) for the type and grid settings that go with these colours.

Examples

```
gfplot_colors("primary")
gfplot_colors(c("model_curve", "comparator_curve"))
gfplot_colors("series")
```

gfplot_families	List the figures gfplot can draw
-----------------	----------------------------------

Description

Returns an index of the available figure families with the category they belong to and how to call them. Families with type = "spec" are driven by `gfplot_spec()` and `gfplot_render()`; families with type = "function" are called directly.

Usage

```
gfplot_families(category = NULL)
```

Arguments

category Optional category to filter by.

Value

A tibble with one row per family: the family name, its title and category, whether it is driven by a specification, the function that draws it, and an example call.

See Also

[gfplot_spec\(\)](#) to build a figure programmatically.

Examples

```
gfplot_families()
gfplot_families("Effect Estimate")
```

gfplot_font_setup	Check which devices can render the figure font
-------------------	--

Description

Figures produced by `gfplot` use Arial. The raster devices supplied by **ragg** and the `quartz` device on macOS resolve system fonts such as Arial directly, so most output needs no preparation at all. The base `pdf()` device keeps its own font table and needs the family registered with it, which **extrafont** does.

Usage

```
gfplot_font_setup(font = "Arial", quiet = FALSE)
```

Arguments

- font Font family to check.
- quiet Suppress the report.

Details

This function reports which routes work in the current session and, when vector PDF output would otherwise be unavailable, registers the font with **extrafont** so that pdf() can use it.

Value

A named logical vector, invisibly, with one entry per device route: raster (**ragg** PNG/TIFF/JPEG), quartz (macOS PDF and screen), and pdf (base pdf()). These are the routes [gfplot_save\(\)](#) chooses between.

See Also

[gfplot_save\(\)](#) to write a figure using a device that can render the font, and [plot_KMCurve\(\)](#) for the font argument.

Examples

```
gfplot_font_setup()
```

gfplot_legend	<i>Legend layout for a set of series labels</i>
---------------	---

Description

Legends in gfplot figures sit under the panel and wrap onto as many rows as the number of series needs, so a wide legend does not squeeze the panel.

Usage

```
gfplot_legend(labels, base_size = 11, hide_single = TRUE)
```

Arguments

- labels Series labels the legend will show.
- base_size Base font size, used to scale the legend keys.
- hide_single Hide the legend when only one series is present, where it would only restate the axis. Set to FALSE when the legend carries information, as with the area under the curve printed by [plot_ROC\(\)](#).

Value

A ggplot2 guide specification, or "none" for a single series when hide_single = TRUE.

Examples

```
gfplot_legend(c("Low risk", "High risk"))
```

gfplot_render

Draw a figure from a specification

Description

Draw a figure from a specification

Usage

```
gfplot_render(spec)
```

Arguments

spec A specification from [gfplot_spec\(\)](#).

Value

A ggplot object, ready to print or pass to [gfplot_save\(\)](#).

See Also

[gfplot_spec\(\)](#)

Examples

```
effects <- data.frame(
  term = c("Age", "Stage III"),
  hr = c(1.02, 1.84),
  lower = c(0.99, 1.51),
  upper = c(1.05, 2.24)
)
p <- gfplot_render(gfplot_spec(
  "forest",
  data = effects, term = "term",
  estimate = "hr", lower = "lower", upper = "upper"
))
```

gfplot_save

Save a figure using a device that can render the requested font

Description

Writing an R figure with Arial is device dependent: the base pdf() device needs the family registered with it, while the **ragg** raster devices and the macOS quartz device resolve system fonts directly. This function chooses a route that can render font for the requested file extension, so a figure is written with the intended typeface instead of silently falling back to a default.

Usage

```
gfplot_save(
  plot,
  filename,
  width = NULL,
  height = NULL,
  dpi = 300,
  size = NULL,
  font = "Arial",
  provenance = FALSE,
  call = NULL,
  ...
)
```

Arguments

plot	A ggplot object, or any object with a print() method that draws it.
filename	Output path. The extension selects the device: .png, .tiff, .jpg, .jpeg, or .pdf.
width, height	Size in inches. Default to 7 by 5, or to the size preset when one is given.
dpi	Resolution for raster output.
size	Journal size preset, used when width and height are not given: "single" for one column, "onehalf" for 1.5 columns, "double" for the full text width, or "slide" for a 16:9 presentation. The journal widths follow the common 85 mm, 114 mm, and 170 mm conventions. Supplying width or height overrides the preset for that dimension.
font	Font family passed to the device unchanged.
provenance	Write a .json file next to the figure recording what produced it: the figure family or call, the package and R versions, the output size and device, and the SHA-256 of the written file. This is what makes a figure in a manuscript traceable to the code that drew it.
call	The call to record in the provenance file. Defaults to the function as it was invoked.
...	Passed on to the device function.

Value

filename, invisibly. The path of the provenance file is attached as the "provenance" attribute when one is written.

See Also

[gfplot_font_setup\(\)](#) to see which devices can render a font.

Examples

```
p <- plot_Boxplot(c(rnorm(20), rnorm(20, 1)), rep(c("A", "B"), each = 20))
path <- tempfile(fileext = ".png")
gfplot_save(p, path, width = 4, height = 3)
unlink(path)
```

gfplot_spec

Build a figure specification

Description

Captures the data and the settings for one figure as a value, validates them, and returns an object that [gfplot_render\(\)](#) turns into a ggplot. Use this when the figure is decided in one place and drawn in another, or when a specification needs to be stored, compared, or passed across a process boundary.

Usage

```
gfplot_spec(family, ...)
```

Arguments

family	Figure family, for example "forest".
...	Family-specific arguments. For "forest": data, term, estimate, lower, and upper, plus the optional settings documented in plot_forest() .

Details

Call [gfplot_families\(\)](#) for the available families and their required arguments.

Value

An object of class `gfplot_spec`.

See Also

[gfplot_render\(\)](#), [plot_forest\(\)](#)

Examples

```
effects <- data.frame(
  term = c("Age", "Stage III"),
  hr = c(1.02, 1.84),
  lower = c(0.99, 1.51),
  upper = c(1.05, 2.24)
)
spec <- gfplot_spec(
  "forest",
  data = effects, term = "term",
  estimate = "hr", lower = "lower", upper = "upper"
)
spec
```

gfplot_theme

House theme for publication figures

Description

The theme every gfplot figure is drawn with: bold axis titles, a thin major grid, a panel border, and a horizontal legend under the panel. Sizes derive from `base_size`, so one number rescales the figure.

Usage

```
gfplot_theme(
  base_size = 11,
  font = "Arial",
  grid = TRUE,
  border = TRUE,
  legend = c("bottom", "right", "top", "left", "none"),
  margin = c(9, 10, 8, 10)
)
```

Arguments

<code>base_size</code>	Base font size in points. Other sizes are derived from it.
<code>font</code>	Font family. Defaults to Arial.
<code>grid</code>	Draw the major grid.
<code>border</code>	Draw the panel border.
<code>legend</code>	Legend position, one of "bottom", "right", "top", "left", or "none".
<code>margin</code>	Plot margin in points, in the order top, right, bottom, left.

Details

This theme is also the session baseline: attaching `gfplot` sets it as the global `ggplot2` theme so that figures drawn by other packages match. Set `options(gfplot.set_theme = FALSE)` before `library(gfplot)` to keep an existing theme instead.

Value

A ggplot2 theme object.

See Also

[gfplot_colors\(\)](#) for the colours that go with it.

Examples

```
library(ggplot2)
p <- ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  gfplot_theme()
```

ggGSEA

Plot a running enrichment score for a ranked gene list

Description

Computes the enrichment score with `DOSE::gseaScores()` and the p-value with `fgsea`, then draws the running score with the leading-edge position.

Usage

```
ggGSEA(input, gs, title = "")
```

Arguments

<code>input</code>	Named numeric vector of ranked statistics.
<code>gs</code>	Gene set: a character vector of gene names.
<code>title</code>	Plot title.

Value

A ggplot object.

`plot_barplot`*Grouped bar plot with optional significance annotations*

Description

Draws the mean of value per group with standard error bars, and adds pairwise significance labels when comparisons is supplied.

Usage

```
plot_barplot(  
  value,  
  group,  
  comparisons = NULL,  
  palette = "house",  
  color = NULL,  
  ylab = "Score",  
  xlab = NULL,  
  title = NULL,  
  font = "Arial",  
  label = "p.signif"  
)
```

Arguments

<code>value</code>	Numeric values to summarise.
<code>group</code>	Grouping vector, one value per observation.
<code>comparisons</code>	Optional list of length-two vectors naming the groups to compare, for example <code>list(c("A", "B"))</code> .
<code>palette</code>	Palette name passed to <code>get_color()</code> .
<code>color</code>	Optional vector of fill colours; overrides palette.
<code>ylab, xlab, title</code>	Axis and plot labels.
<code>font</code>	Font family used in the plot.
<code>label</code>	Significance label style passed to <code>ggpubr::stat_compare_means()</code> .

Details

This function replaced the 0.1.0 `plot_barplot()`, which took no arguments and could not be called.

Value

A ggplot object.

Examples

```

set.seed(1)
p <- plot_barplot(
  value = c(rnorm(20), rnorm(20, 1)),
  group = rep(c("A", "B"), each = 20)
)

if (requireNamespace("ggpubr", quietly = TRUE)) {
  p <- plot_barplot(
    value = c(rnorm(20), rnorm(20, 1), rnorm(20, 2)),
    group = rep(c("A", "B", "C"), each = 20),
    comparisons = list(c("A", "B"), c("B", "C"))
  )
}

```

plot_Boxplot

*Boxplot with optional pairwise significance annotations***Description**

Boxplot with optional pairwise significance annotations

Usage

```

plot_Boxplot(
  value,
  label,
  palette = "house",
  title = NULL,
  ylab = "Expression",
  font = "Arial"
)

```

Arguments

value	Numeric values.
label	Grouping vector.
palette	Palette name passed to <code>get_color()</code> .
title	Plot title.
ylab	Y-axis label.
font	Font family used in the plot.

Value

A ggplot object.

Examples

```
set.seed(1)
p <- plot_Boxplot(c(rnorm(20), rnorm(20, 1)), rep(c("A", "B"), each = 20))
```

plot_calibration	<i>Calibration curve</i>
------------------	--------------------------

Description

Predicted probability against the observed proportion of events. A model whose points sit on the diagonal predicts the level it claims; one above the diagonal under-predicts risk and one below over-predicts.

Usage

```
plot_calibration(
  probability,
  outcome,
  positive = NULL,
  bins = 10,
  smooth = FALSE,
  show_ci = TRUE,
  palette = "house",
  ci_level = 0.95,
  xlab = "Predicted probability",
  ylab = "Observed proportion",
  title = NULL,
  font = "Arial"
)
```

Arguments

probability	Predicted probabilities, or a matrix or data frame with one column per model.
outcome	Binary outcome, one value per observation.
positive	Value of outcome that represents the event.
bins	Number of groups to divide the predictions into. Quantile bins are used, so every point rests on a similar number of observations.
smooth	Draw a loess curve through the points instead of connecting them.
show_ci	Draw binomial confidence intervals for each point.
palette	Palette name passed to get_color() .
ci_level	Confidence level of the intervals.
xlab, ylab, title	Axis and plot labels.
font	Font family used in the plot.

Value

A ggplot object.

See Also

[plot_ROC\(\)](#) for discrimination, [plot_decision_curve\(\)](#) for whether acting on the predictions helps.

Examples

```
set.seed(1)
outcome <- rbinom(200, 1, 0.4)
probability <- plogis(rnorm(200) + outcome * 0.5)
p <- plot_calibration(probability, outcome)
```

plot_confusion	<i>Confusion matrix</i>
----------------	-------------------------

Description

Predicted against actual class, with the count in each cell. Percentages are given per actual class, which is the direction that shows sensitivity and specificity; overall percentages would hide a class the model misses.

Usage

```
plot_confusion(
  predicted = NULL,
  actual = NULL,
  predicted_labels = NULL,
  actual_labels = NULL,
  positive = NULL,
  show = c("count", "percent", "both"),
  digits = 1,
  title = NULL,
  font = "Arial"
)
```

Arguments

predicted, predicted_labels
Predicted class labels.

actual, actual_labels
Actual class labels. actual may also be a table or matrix of counts, in which case the labels are taken from its dimnames.

positive	Value of the outcome that counts as the positive class, used to order and label the axes. With positive supplied, the two classes are shown as "Negative" and "Positive"; pass predicted_labels and actual_labels to name them differently.
show	Which numbers to print: "count", "percent" (by actual class), or "both".
digits	Significant digits for percentages.
title	Plot title.
font	Font family used in the plot.

Value

A ggplot object.

Examples

```
set.seed(1)
actual <- rbinom(200, 1, 0.4)
predicted <- ifelse(runif(200) < 0.8, actual, 1 - actual)
p <- plot_confusion(predicted, actual, positive = 1)
```

plot_cor

Plot the correlation between two variables

Description

Plot the correlation between two variables

Usage

```
plot_cor(x, y, groups = NULL, xlab = NULL, ylab = NULL, legend.pos = "top")
```

Arguments

x, y	Numeric vectors.
groups	Optional grouping vector used to colour the points.
xlab, ylab	Axis labels.
legend.pos	Legend position.

Value

A ggplot object.

Examples

```
p <- plot_cor(iris$Sepal.Length, iris$Sepal.Width)
```

plot_cumulative_incidence

Cumulative incidence of competing events

Description

The probability of each event type over follow-up, in the presence of competing events. When a patient can experience one of several mutually exclusive events, a Kaplan-Meier estimate of any single one is biased upward, because patients who had a competing event are still counted as being at risk of the event of interest. The cumulative incidence function accounts for them.

Usage

```
plot_cumulative_incidence(
  time,
  status,
  group = NULL,
  event = NULL,
  conf_int = FALSE,
  palette = "house",
  xlab = "Follow up",
  ylab = "Cumulative incidence",
  title = NULL,
  legend_title = NULL,
  font = "Arial"
)
```

Arguments

time	Follow-up times.
status	Event type per patient. The first level is the event of interest, 0 is censoring, and any further levels are competing events. A factor with a 0 level, a numeric code vector, or a character vector are all accepted.
group	Optional grouping vector, giving one panel per group.
event	Optional event type to plot. By default every event type is drawn.
conf_int	Draw pointwise confidence bands.
palette	Palette name passed to get_color() .
xlab, ylab, title	Axis and plot labels.
legend_title	Legend title.
font	Font family used in the plot.

Value

A ggplot object.

See Also

[plot_KMCurve\(\)](#) for a single event where no competing risk applies.

Examples

```
set.seed(1)
time <- rexp(200)
status <- sample(c(0, 1, 2), 200, replace = TRUE, prob = c(0.4, 0.4, 0.2))
p <- plot_cumulative_incidence(time, status, group = rep(c("A", "B"), 100))
```

plot_decision_curve *Decision curve*

Description

Net benefit across decision thresholds. Net benefit weights the true positives a model finds against the false positives it costs, at the threshold probability a clinician would use to act. It is the figure that answers whether using the model is better than treating everyone or no one.

Usage

```
plot_decision_curve(
  probability,
  outcome,
  positive = NULL,
  thresholds = seq(0.01, 0.5, by = 0.01),
  treat_all = TRUE,
  treat_none = TRUE,
  palette = "house",
  xlab = "Threshold probability",
  ylab = "Net benefit",
  title = NULL,
  font = "Arial"
)
```

Arguments

probability	Predicted probabilities, or a matrix or data frame with one column per model.
outcome	Binary outcome, one value per observation.
positive	Value of outcome that represents the event.
thresholds	Decision thresholds to evaluate. Defaults to 0.01 to 0.5, which covers the range used in most clinical decisions.
treat_all	Draw the reference curve for treating every patient.
treat_none	Draw the reference line for treating no patient.
palette	Palette name passed to get_color() .

xlab, ylab, title Axis and plot labels.

font Font family used in the plot.

Value

A ggplot object.

See Also

[plot_calibration\(\)](#) for whether the predicted levels are honest.

Examples

```
set.seed(1)
outcome <- rbinom(300, 1, 0.3)
probability <- plogis(rnorm(300) + outcome)
p <- plot_decision_curve(probability, outcome)
```

plot_embedding	<i>Grouped embedding scatter</i>
----------------	----------------------------------

Description

Plot samples in a two-dimensional embedding, coloured by group. The embedding can be supplied already computed, or computed here from a feature matrix with [stats::prcomp\(\)](#), **Rtsne**, or **umap**.

Usage

```
plot_embedding(
  data,
  groups,
  method = c("pca", "tsne", "umap", "none"),
  perplexity = 30,
  seed = NULL,
  ellipse = FALSE,
  label_centres = FALSE,
  palette = "house",
  title = NULL,
  xlab = NULL,
  ylab = NULL,
  font = "Arial"
)
```

Arguments

data	A feature matrix with samples in rows, or a two-column matrix of coordinates when method = "none".
groups	Group labels, one per sample.
method	Embedding to compute: "pca", "tsne", "umap", or "none" when data already holds coordinates.
perplexity	Perplexity for t-SNE. Values much larger than the number of samples are rejected by the back end.
seed	Optional seed for the stochastic embeddings.
ellipse	Draw a confidence ellipse per group.
label_centres	Label each group at the centre of its points.
palette	Palette name passed to <code>get_color()</code> .
title, xlab, ylab	Axis and plot labels.
font	Font family used in the plot.

Details

`plot_PCA()` and `plot_UMAP()` are thin wrappers around this function, so a figure keeps the same look whichever embedding it shows.

Value

A ggplot object.

See Also

`plot_PCA()`, `plot_UMAP()`

Examples

```
set.seed(1)
data <- matrix(rnorm(200), nrow = 20)
groups <- rep(c("A", "B"), each = 10)
p <- plot_embedding(data, groups, method = "pca")
```

plot_forest	<i>Forest plot of effect estimates</i>
-------------	--

Description

Draws point estimates with confidence intervals on a horizontal axis, with the numbers printed alongside. This is the standard display for hazard ratios, odds ratios, and mean differences, and it reads a table straight from `clinstats::cox_table()` or `clinstats::logistic_table()`.

Usage

```
plot_forest(
  data,
  term,
  estimate,
  lower,
  upper,
  p = NULL,
  group = NULL,
  label = NULL,
  log_scale = FALSE,
  ref_line = NULL,
  estimate_label = "Estimate (95% CI)",
  p_label = "P",
  show_table = TRUE,
  digits = 2,
  xlab = NULL,
  title = NULL,
  font = "Arial"
)
```

Arguments

<code>data</code>	Data frame holding the estimates.
<code>term</code>	Column of row labels, one per estimate.
<code>estimate</code>	Column of point estimates.
<code>lower, upper</code>	Columns of the confidence limits.
<code>p</code>	Optional column of p-values, shown as a right-hand column.
<code>group</code>	Optional column naming the section each estimate belongs to. Sections are drawn as facets with a shared axis.
<code>label</code>	Optional column overriding <code>term</code> for display, for example a formatted "Age, per year" instead of a variable name.
<code>log_scale</code>	Plot the axis on a log scale, which is what ratios need so that a halving and a doubling are the same distance.
<code>ref_line</code>	Value of the reference line. Defaults to 1 when <code>log_scale = TRUE</code> and 0 otherwise.
<code>estimate_label</code>	Column heading for the estimate table.
<code>p_label</code>	Column heading for the p-value column.
<code>show_table</code>	Print the estimate and p-value columns.
<code>digits</code>	Significant digits for the printed numbers.
<code>xlab, title</code>	Axis and plot labels.
<code>font</code>	Font family used in the plot.

Value

A ggplot object, or a composed panel when `show_table = TRUE`, as with the risk table of `plot_KMCurve()`. Both print and save the same way.

See Also

`gfplot_spec()` to build the same figure as a value. The companion package `clinstats` produces the estimates this draws.

Examples

```
effects <- data.frame(
  term = c("Age, per year", "Stage III vs II", "MSI-high"),
  hr = c(1.02, 1.84, 0.62),
  lower = c(0.99, 1.51, 0.44),
  upper = c(1.05, 2.24, 0.88),
  p = c(0.21, 4.1e-07, 0.008)
)
p <- plot_forest(
  effects,
  term = "term", estimate = "hr", lower = "lower", upper = "upper",
  p = "p", log_scale = TRUE
)

# Straight from a clinstats table

if (requireNamespace("clinstats", quietly = TRUE)) {
  tbl <- clinstats::cox_table(
    clinstats::clin_crc,
    time = "rfs.delay", event = "rfs.event",
    factors = c("sex", "age", "tnm.stage"), multivariable = "none"
  )
  p <- plot_forest(
    tbl,
    term = "term", estimate = "hr_univariable",
    lower = "univ_ci_lower", upper = "univ_ci_upper", p = "univ_p",
    log_scale = TRUE
  )
}
```

plot_GO

Dot plot of enriched terms

Description

Plots the most significant terms with gene ratio on the x-axis, term on the y-axis, point size by gene count, and colour by significance.

Usage

```
plot_GO(gsea, n = 20, font = "Arial")
```

Arguments

gsea	Result table with the columns p_value, term_name, intersection_size, and query_size.
n	Maximum number of terms to display.
font	Font family used in the plot.

Value

A ggplot object.

plot_heatmap	<i>Heatmap of a matrix</i>
--------------	----------------------------

Description

Draws the values of a matrix as tiles with a colourbar. This is the display for correlation matrices, signature scores by group, and any rectangular table of values where the pattern matters more than the numbers.

Usage

```
plot_heatmap(
  matrix,
  row_order = NULL,
  column_order = NULL,
  values = FALSE,
  digits = 2,
  midpoint = NULL,
  diverging = FALSE,
  low = NULL,
  mid = NULL,
  high = NULL,
  legend_label = "Value",
  title = NULL,
  font = "Arial"
)
```

Arguments

matrix	Numeric matrix, or a data frame whose columns are numeric.
row_order, column_order	Optional ordering of rows and columns. Use "cluster" to order by hierarchical clustering, "value" to order by the first principal component, or supply a character vector of names.
values	Print the numeric value in each tile.
digits	Significant digits for the printed values.
midpoint	Value mapped to the middle of the colourbar. Defaults to the median, which centres a diverging scale on a typical value.
diverging	Use a diverging scale, for values that have a meaningful centre such as a correlation.
low, mid, high	Colours for the low, middle, and high ends.
legend_label	Title of the colourbar.
title	Plot title.
font	Font family used in the plot.

Value

A ggplot object.

Examples

```
set.seed(1)
m <- cor(matrix(rnorm(60), nrow = 20))
diag(m) <- NA
p <- plot_heatmap(m, diverging = TRUE, midpoint = 0,
  legend_label = "Correlation")
```

plot_immune	<i>Radar chart of immune-cell infiltration by group</i>
-------------	---

Description

Radar chart of immune-cell infiltration by group

Usage

```
plot_immune(res, group, title = "", legend.position = "left")
```

Arguments

<code>res</code>	Immune-cell estimates with cell types in the first column and samples in the remaining columns.
<code>group</code>	Named group labels, indexed by sample.
<code>title</code>	Plot title.
<code>legend.position</code>	Legend position.

Value

A ggplot object.

<code>plot_KMCurve</code>	<i>Plot a Kaplan-Meier curve with risk table and hazard ratio</i>
---------------------------	---

Description

Draws survival curves for two or more groups, with an optional number-at-risk table, the log-rank p-value, and, for two groups, the hazard ratio from a univariable Cox model.

Usage

```
plot_KMCurve(
  clinical,
  labels,
  limit = NULL,
  annot = NULL,
  color = NULL,
  font = "Arial",
  xlab = "Follow up",
  ylab = "Survival Probability",
  title = NULL,
  legend.pos = "top",
  palette = "house",
  risk.table = TRUE,
  risk.table.ratio = 0.4,
  anno.pos = "bottom",
  anno.x.shift = 0.5
)
```

Arguments

<code>clinical</code>	A survival object created by <code>survival::Surv()</code> .
<code>labels</code>	A vector of group labels, one per patient.
<code>limit</code>	Optional follow-up limit; later observations are censored at the limit.

annot	Optional extra annotation added to the plot.
color	Optional vector of colours. A named vector also sets the group order.
font	Font family used in the plot.
xlab, ylab, title	Axis and plot labels.
legend.pos	Legend position.
palette	Palette name passed to <code>get_color()</code> .
risk.table	Draw the number-at-risk table.
risk.table.ratio	Relative height of the risk table.
anno.pos	Either "bottom" or "top"; where the annotations are placed.
anno.x.shift	Horizontal position of the annotations when <code>anno.pos = "top"</code> .

Value

A ggplot object, or a cowplot grid when `risk.table = TRUE`.

Examples

```
if (requireNamespace("survminer", quietly = TRUE)) {
  fit_data <- survival::lung
  p <- plot_KMCurve(
    survival::Surv(fit_data$time, fit_data$status == 2),
    factor(fit_data$sex),
    risk.table = FALSE
  )
}
```

plot_lasso

Plot lasso coefficient paths

Description

Draws the coefficient paths of a fitted `glmnet` model against the log lambda sequence and labels the coefficients that remain non-zero at the selected value of `s`.

Usage

```
plot_lasso(fit, s, font = "Arial")
```

Arguments

fit	A fitted model from <code>glmnet::glmnet()</code> .
s	Selected value of the penalty parameter.
font	Font family used in the plot.

Value

A ggplot object.

Examples

```
if (requireNamespace("glmnet", quietly = TRUE)) {
  set.seed(1)
  x <- matrix(rnorm(100 * 5), nrow = 100)
  y <- rnorm(100) + x[, 1]
  p <- plot_lasso(glmnet::glmnet(x, y), s = 0.05)
}
```

plot_MulROC

Plot several ROC curves from separate datasets

Description

Each element of scores is matched with the corresponding element of labels, which allows the curves to come from different cohorts or endpoints rather than from different markers of the same cohort.

Usage

```
plot_MulROC(
  scores,
  labels,
  palette = "house",
  color = NULL,
  legend.pos = "bottom",
  title = NULL,
  font = "Arial",
  percent.style = FALSE
)
```

Arguments

scores	A named list of numeric marker vectors.
labels	A list of binary outcomes, one per element of scores.
palette	Palette name passed to get_color() .
color	Optional vector of colours; overrides palette.
legend.pos	Legend position: a position such as "bottom", or an x/y coordinate pair in panel units for a legend inside the panel.
title	Plot title.
font	Font family used in the plot.
percent.style	Label the axes as percentages.

Value

A ggplot object.

Examples

```
set.seed(1)
scores <- list(cohortA = rnorm(60), cohortB = rnorm(60))
labels <- list(rbinom(60, 1, 0.5), rbinom(60, 1, 0.5))
p <- plot_Mu1ROC(scores, labels)
```

plot_PCA

Principal component analysis plot

Description

Principal component analysis plot

Usage

```
plot_PCA(
  data,
  labs,
  title = "Evaluate the batch effect between groups",
  palette = "house",
  font = "Arial",
  ...
)
```

Arguments

data	Matrix or data frame with samples in rows and features in columns.
labs	Group labels, one per sample.
title	Plot title.
palette	Palette name passed to get_color() .
font	Font family used in the plot.
...	Passed to plot_embedding() , for example ellipse, seed, or label_centres.

Value

A ggplot object.

Examples

```
set.seed(1)
p <- plot_PCA(matrix(rnorm(200), nrow = 20), rep(c("A", "B"), each = 10))
```

plot_pr_curve	<i>Precision-recall curve</i>
---------------	-------------------------------

Description

Precision against recall for a binary outcome. On an imbalanced outcome the ROC curve can look encouraging while precision stays low, which is what this plot exposes: it is the recommended companion when the event is rare.

Usage

```
plot_pr_curve(  
  scores,  
  labels,  
  positive = NULL,  
  palette = "house",  
  title = NULL,  
  font = "Arial",  
  show_auc = TRUE  
)
```

Arguments

scores	Numeric vector, or a matrix or data frame with one column per marker.
labels	Binary outcome, one value per observation.
positive	Percentile of the scores above which the positive class is expected. Used only to draw the no-skill reference line, which is the prevalence of the positive class.
palette	Palette name passed to get_color() .
title	Plot title.
font	Font family used in the plot.
show_auc	Print the area under the curve, computed by the trapezoid rule, in the legend.

Value

A ggplot object.

See Also

[plot_ROC\(\)](#) for discrimination, [plot_calibration\(\)](#) for the level of the predictions.

Examples

```
set.seed(1)  
outcome <- rbinom(200, 1, 0.15)  
score <- outcome * 0.8 + rnorm(200)  
p <- plot_pr_curve(score, outcome)
```

plot_RiskScore	<i>Plot a risk score ordered by patient</i>
----------------	---

Description

Plot a risk score ordered by patient

Usage

```
plot_RiskScore(  
  rs,  
  event,  
  legend.position = c(0.2, 0.8),  
  palette = "house",  
  color = NULL,  
  font = "Arial"  
)
```

Arguments

rs	Numeric risk scores. Names, when present, are used as patient identifiers.
event	Event indicator used to fill the bars.
legend.position	Legend position.
palette	Palette name passed to get_color() .
color	Optional vector of fill colours; overrides palette.
font	Font family used in the plot.

Value

A ggplot object.

Examples

```
set.seed(1)  
p <- plot_RiskScore(rnorm(30), rbinom(30, 1, 0.4))
```

plot_ROC*Plot ROC curves for one or more markers*

Description

Plots receiver operating characteristic curves with the area under the curve and its confidence interval in the legend.

Usage

```
plot_ROC(  
  scores,  
  labels,  
  force05 = FALSE,  
  palette = "house",  
  legend.pos = "bottom",  
  title = NULL,  
  font = "Arial",  
  percent.style = FALSE  
)
```

Arguments

scores	Numeric vector, or a matrix or data frame with one column per marker.
labels	Binary outcome, one value per observation.
force05	Flip markers whose area under the curve is below 0.5 so that every curve is plotted above the diagonal.
palette	Palette name passed to get_color() .
legend.pos	Legend position: a position such as "bottom", or an x/y coordinate pair in panel units for a legend inside the panel.
title	Plot title.
font	Font family used in the plot.
percent.style	Label the axes as percentages.

Value

A ggplot object.

Examples

```
set.seed(1)  
scores <- cbind(marker1 = rnorm(80), marker2 = rnorm(80))  
labels <- rep(c(0, 1), each = 40)  
p <- plot_ROC(scores, labels)
```

plot_TimeROC	<i>Plot time-dependent ROC curves</i>
--------------	---------------------------------------

Description

Plots survival ROC curves at one or more time points using the Kaplan-Meier method of the survivalROC package.

Usage

```
plot_TimeROC(
  scores,
  survival,
  time_points,
  groups,
  palette = "house",
  legend.pos = "bottom",
  title = NULL,
  font = "Arial",
  percent.style = FALSE
)
```

Arguments

scores	Numeric marker values.
survival	A two-column survival object.
time_points	Time points at which the curves are evaluated.
groups	Labels for the curves, one per time point.
palette	Palette name passed to get_color() .
legend.pos	Legend position: a position such as "bottom", or an x/y coordinate pair in panel units for a legend inside the panel.
title	Plot title.
font	Font family used in the plot.
percent.style	Label the axes as percentages.

Value

A ggplot object.

Examples

```
if (requireNamespace("survivalROC", quietly = TRUE)) {
  set.seed(1)
  surv <- survival::Surv(rexp(100), rbinom(100, 1, 0.6))
  p <- plot_TimeROC(rnorm(100), surv, c(1, 2), c("year1", "year2"))
}
```

plot_tsne*t-SNE projection of samples coloured by group*

Description

t-SNE projection of samples coloured by group

Usage

```
plot_tsne(  
  data,  
  labs,  
  title = "Evaluate the batch effect between groups",  
  perplexity = 30,  
  seed = 1,  
  palette = "house",  
  font = "Arial",  
  ...  
)
```

Arguments

data	Matrix or data frame with samples in rows and features in columns.
labs	Group labels, one per sample.
title	Plot title.
perplexity	Perplexity for the t-SNE embedding.
seed	Optional seed; t-SNE is stochastic, so a seed makes a figure reproducible.
palette	Palette name passed to get_color() .
font	Font family used in the plot.
...	Passed to plot_embedding() , for example ellipse, seed, or label_centres.

Value

A ggplot object.

Examples

```
if (requireNamespace("Rtsne", quietly = TRUE)) {  
  set.seed(1)  
  p <- plot_tsne(matrix(rnorm(200), nrow = 20), rep(c("A", "B"), each = 10))  
}
```

plot_UMAP	<i>UMAP projection plot</i>
-----------	-----------------------------

Description

UMAP projection plot

Usage

```
plot_UMAP(  
  data,  
  labs,  
  title = "Evaluate the batch effect between groups",  
  palette = "house",  
  font = "Arial",  
  ...  
)
```

Arguments

data	Matrix or data frame with samples in rows and features in columns.
labs	Group labels, one per sample.
title	Plot title.
palette	Palette name passed to get_color() .
font	Font family used in the plot.
...	Passed to plot_embedding() .

Value

A ggplot object.

Examples

```
if (requireNamespace("umap", quietly = TRUE)) {  
  set.seed(1)  
  p <- plot_UMAP(matrix(rnorm(200), nrow = 20), rep(c("A", "B"), each = 10))  
}
```

plot_violin

*Compare distributions across groups***Description**

Violin plots with the individual observations overlaid, which shows both the shape of each distribution and how much data stands behind it. A boxplot alone hides a bimodal group and a group with four samples.

Usage

```
plot_violin(
  value,
  group,
  order = NULL,
  show = c("violin", "box", "both"),
  points = TRUE,
  points_maximum = 300,
  show_n = TRUE,
  significance = FALSE,
  palette = "house",
  ylab = NULL,
  xlab = NULL,
  title = NULL,
  font = "Arial"
)
```

Arguments

value	Numeric values.
group	Grouping vector, one value per observation.
order	Group order. Defaults to the order of the levels, or of first appearance for a plain vector.
show	Which to draw: "violin", "box", or "both".
points	Draw the individual observations.
points_maximum	Draw the observations only when there are at most this many, since a violin over thousands of points is unreadable. Use Inf to always draw them.
show_n	Print the number of observations under each group.
significance	Add pairwise Wilcoxon comparisons. TRUE compares every pair; a list such as <code>list(c("A", "B"))</code> compares those pairs only.
palette	Palette name passed to <code>get_color()</code> .
ylab, xlab, title	Axis and plot labels.
font	Font family used in the plot.

Value

A ggplot object.

See Also

`plot_Boxplot()` for the box-only version.

Examples

```
set.seed(1)
p <- plot_violin(c(rnorm(40), rnorm(40, 1)), rep(c("A", "B"), each = 40))
```

plot_volcano	<i>Volcano plot</i>
--------------	---------------------

Description

Effect size against significance, with the calling thresholds drawn. This is the standard first look at a differential analysis: it separates the features that moved a lot from the ones that moved reliably, and shows how many of each the thresholds produce.

Usage

```
plot_volcano(
  effect,
  p_value,
  label = NULL,
  effect_threshold = 1,
  p_threshold = 0.05,
  label_top = 10,
  palette = "house",
  repel = TRUE,
  xlab = "Effect size",
  ylab = expression(-log[10] ~ italic(P)),
  title = NULL,
  font = "Arial"
)
```

Arguments

- | | |
|-------------------------------|--|
| effect | Numeric effect sizes, typically a log fold change. |
| p_value | Significance values on the same scale as p_threshold. |
| label | Feature labels. |
| effect_threshold, p_threshold | Calling thresholds. Features past both are labelled as significant. |
| label_top | Label the most significant features instead of every significant one. Set to 0 to label none, or Inf to label all. |

palette	Palette name passed to <code>get_color()</code> .
repel	Repel the labels so they do not overlap.
xlab, ylab, title	Axis and plot labels.
font	Font family used in the plot.

Value

A ggplot object.

Examples

```
set.seed(1)
effect <- rnorm(2000)
p_value <- runif(2000)^2
p <- plot_volcano(effect, p_value, paste0("gene", seq_along(effect)))
```

plot_waterfall	<i>Waterfall plot of response</i>
----------------	-----------------------------------

Description

Patients ranked by response, with the bar height showing how far each departed from baseline and the fill showing the response category. This is the standard display of response in a single-arm trial.

Usage

```
plot_waterfall(
  response,
  category = NULL,
  patient = NULL,
  label_top = 0,
  thresholds = c(`Partial response` = -30, `Stable disease` = 20),
  threshold_labels = TRUE,
  show_patients = FALSE,
  palette = "house",
  xlab = NULL,
  ylab = "Change from baseline (%)",
  title = NULL,
  font = "Arial"
)
```

Arguments

response	Numeric response values, for example a percentage change from baseline. NA values are dropped.
category	Optional response category per patient, used to fill the bars. Defaults to the RECIST-style cut-offs, where -30 and 20 percent separate partial response, stable disease, and progression.
patient	Optional patient labels, used on the x axis when few enough bars are drawn to read them.
label_top	Label this many of the largest and smallest bars, which is what a reader wants from a dense waterfall.
thresholds	Cut-offs between response categories, as a named vector such as <code>c("Partial response" = -30, "Stable disease" = 20)</code> . Values at or below the first are partial response and values above the last are progression.
threshold_labels	Print the cut-off lines and their values.
show_patients	Print patient labels under the bars.
palette	Palette name passed to <code>get_color()</code> .
xlab, ylab, title	Axis and plot labels.
font	Font family used in the plot.

Value

A ggplot object.

Examples

```
set.seed(1)
response <- c(rnorm(40, -10, 20), rnorm(10, 40, 15))
p <- plot_waterfall(response)
```

viewGSEA

Plot a running enrichment score for an HTSanalyzeR2 result

Description

Updated version of the viewGSEA() helper distributed with HTSanalyzeR2.

Usage

```
viewGSEA(object, gscName, gsName, title = "")
```

Arguments

<code>object</code>	An HTSanalyzeR2 result object.
<code>gscName</code>	Name of the gene set collection.
<code>gsName</code>	Name of the gene set.
<code>title</code>	Plot title.

Value

A ggplot object.

Index

generate_time_event, 3
get_color, 3
get_color(), 4, 12–14, 17, 18, 20, 26–35, 37, 38
gfplot_colors, 4
gfplot_colors(), 3, 11
gfplot_families, 5
gfplot_font_setup, 5
gfplot_font_setup(), 9
gfplot_legend, 6
gfplot_render, 7
gfplot_render(), 9
gfplot_save, 8
gfplot_save(), 6, 7
gfplot_spec, 9
gfplot_spec(), 5, 7, 22
gfplot_theme, 10
gfplot_theme(), 4
ggGSEA, 11
ggpubr::stat_compare_means(), 12
glmnet::glmnet(), 26

plot_barplot, 12
plot_Boxplot, 13
plot_Boxplot(), 36
plot_calibration, 14
plot_calibration(), 19, 29
plot_confusion, 15
plot_cor, 16
plot_cumulative_incidence, 17
plot_decision_curve, 18
plot_decision_curve(), 15
plot_embedding, 19
plot_embedding(), 28, 33, 34
plot_forest, 20
plot_forest(), 9
plot_GO, 22
plot_heatmap, 23
plot_immune, 24
plot_KMCurve, 25
plot_KMCurve(), 6, 18, 22
plot_lasso, 26
plot_MulROC, 27
plot_PCA, 28
plot_PCA(), 20
plot_pr_curve, 29
plot_RiskScore, 30
plot_ROC, 31
plot_ROC(), 6, 15, 29
plot_TimeROC, 32
plot_tsne, 33
plot_UMAP, 34
plot_UMAP(), 20
plot_violin, 35
plot_volcano, 36
plot_waterfall, 37

stats::prcomp(), 19
survival::Surv(), 25

viewGSEA, 38