Package: ggcolonviz (via r-universe)

October 16, 2026

Title A 'ggplot2' Extension for Visualizing the Colon

Version 0.0.6

SystemRequirements GDAL (>= 2.0.1), GEOS (>= 3.4.0), PROJ (>= 4.8.0), sqlite3

Description Provides a 'ggplot2' extension for creating clear, customizable visualizations of the colon. Includes geometry and label layers for plotting colon anatomy and overlaying related data, along with helpers for converting colon segments to 'sf' polygons and generating heatmap grids across the colon geometry.

License Apache License (>= 2)

Copyright file inst/NOTICE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

URL https://github.com/Sanofi-Public/ggcolonviz

BugReports https://github.com/Sanofi-Public/ggcolonviz/issues

Config/testthat/edition 3

Depends R (>= 4.2.0)

Imports ggplot2, sf, dplyr

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository https://sanofi-public.r-universe.dev

Date/Publication 2026-09-29 19:06:31 UTC

RemoteUrl https://github.com/sanofi-public/ggcolonviz

RemoteRef HEAD

RemoteSha 991e87500b1881414cab0ad8997e71769e462911

Contents

ggcolonviz-package 2
colon_heatmap_grid 2
colon_sf 3
geom_colon 4
geom_colon_label 5
list_colon_variants 6
register_colon_variant 7

Index **8**

`ggcolonviz-package` `ggcolonviz` *Package*

Description

`ggcolonviz` implements a ggplot2 extension for visualizing the colon.

Author(s)

Maintainer: Yoni Sidi <yoni.sidi@sanofi.com>

Other contributors:

- Sanofi [copyright holder]

See Also

Useful links:

- https://github.com/Sanofi-Public/ggcolonviz
- Report bugs at https://github.com/Sanofi-Public/ggcolonviz/issues

---

`colon_heatmap_grid` *Colon Heatmap Grid*

Description

This function creates a grid of points within the colon geometry, which can be used to generate heatmaps or other spatial visualizations. The grid is created by intersecting a dense grid of points with the colon geometry, resulting in a set of points that lie within the colon structure.

Usage

```
colon_heatmap_grid(grid_res = 500, variant = "standard")
```

Arguments

`grid_res`	The resolution of the grid, which determines how many points are generated. A higher resolution will create a denser grid, which can lead to smoother heatmaps but may also increase computation time.
`variant`	A character string specifying the variant of the colon blueprint to use.

Details

The function first creates a dense grid of points based on the specified resolution, then intersects this grid with the colon geometry to retain only the points that lie within the colon. The resulting sf object contains the coordinates of the points that can be used for further analysis or visualization.

Value

An sf object containing the points that lie within the colon geometry, which can be used for heatmap generation or other spatial analyses.

`colon_sf` *Colon Geometry as sf Object*

Description

This function converts the colon blueprint into an sf object, which can be used for spatial operations and visualizations. It groups the blueprint by colon parts and creates polygon geometries for each part.

Usage

```
colon_sf(variant = "standard")
```

Arguments

`variant`	A character string specifying the variant of the colon blueprint to use.

Details

The resulting sf object contains the geometry of the colon parts, which can be used for various spatial analyses or visualizations, such as creating heatmaps or overlaying additional data on the colon structure.

Value

An sf object with polygon geometries for each part of the colon.

geom_colon *Colon Geometry Layer for ggplot2*

Description

This function defines a custom ggplot2 layer that allows users to plot the colon geometry based on the predefined blueprint. It takes in aesthetics for x, y, and part, and uses the StatColon ggproto object to compute the necessary data for plotting the colon structure. The resulting layer can be used in conjunction with other ggplot2 layers to create detailed visualizations of the colon.

Usage

```
geom_colon(
  mapping = NULL,
  data = NULL,
  variant = "standard",
  show.legend = NA,
  ...
)
```

Arguments

`mapping`	Set of aesthetic mappings created by `aes()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping.
`data`	The data to be displayed in this layer. There are three options: If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`. A `data.frame`, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `fortify()` for which variables will be created. A `function` will be called with a single argument, the plot data. The return value must be a `data.frame`, and will be used as the layer data. A `function` can be created from a `formula` (e.g. `~ head(.x, 10)`).
`variant`	A character string specifying the variant of the colon blueprint to use.
`show.legend`	logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use `TRUE`. If `NA`, all levels are shown in legend, but unobserved levels are omitted.
`...`	Additional parameters passed to the layer, such as fill, color, size, etc.

Details

The geom_colon function creates a layer that can be added to a ggplot2 plot. It requires the aesthetics x, y, and part to be specified, which correspond to the position and the specific part of the colon to be plotted.

The variants can be defined in the colon_variants list, which should contain different blueprints for the colon geometry.

Value

A ggplot2 layer that can be added to a plot to visualize the colon geometry.

---

geom_colon_label *Colon Label Layer for ggplot2*

Description

This function defines a custom ggplot2 layer that allows users to add labels to the colon parts based on the predefined blueprint. It takes in aesthetics for x, y, and part, and uses the StatColonLabels ggproto object to compute the necessary data for positioning and labeling the colon parts. The resulting layer can be used in conjunction with other ggplot2 layers to create detailed visualizations of the colon with informative labels.

Usage

```
geom_colon_label(
  mapping = NULL,
  data = NULL,
  show.legend = NA,
  variant = "standard",
  ...
)
```

Arguments

`mapping` Set of aesthetic mappings created by `aes()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping.

`data` The data to be displayed in this layer. There are three options:

If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`.

A `data.frame`, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `fortify()` for which variables will be created.

A `function` will be called with a single argument, the plot data. The return value must be a `data.frame`, and will be used as the layer data. A `function` can be created from a `formula` (e.g. `~ head(.x, 10)`).

`show.legend`	logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use `TRUE`. If `NA`, all levels are shown in legend, but unobserved levels are omitted.
`variant`	A character string specifying the variant of the colon blueprint to use.
`...`	Additional parameters passed to the layer, such as fill, color, size, etc.

Details

The geom_colon_label function creates a layer that can be added to a ggplot2 plot. It requires the aesthetics x, y, and part to be specified, which correspond to the position and the specific part of the colon to be labeled.

The variants can be defined in the colon_variants list, which should contain different blueprints for the colon geometry.#'

Value

A ggplot2 layer that can be added to a plot to visualize labels on the colon parts.

`list_colon_variants` *List Registered Colon Variant Names*

Description

Returns the names of all variants currently registered in the package registry, including the built-in `"standard"` and `"sescd"` variants plus any added via `register_colon_variant()`.

Usage

```
list_colon_variants()
```

Value

A character vector of variant names.

Examples

```
list_colon_variants()
```

register_colon_variant

Register a Custom Colon Variant

Description

Adds a new named blueprint to the package's internal variant registry so it can be referenced by `variant` arguments in `geom_colon()`, `geom_colon_label()`, `colon_sf()`, and `colon_heatmap_grid()`.

Usage

```
register_colon_variant(name, data)
```

Arguments

`name`	A non-empty character string naming the new variant. Existing variants (including `"standard"` and `"sescd"`) will be overwritten if the same name is supplied.
`data`	A data frame with at least three columns: `dx` Numeric. Relative x-coordinate of each polygon vertex. `dy` Numeric. Relative y-coordinate of each polygon vertex. `part` Character. Segment name each vertex belongs to. Each unique `part` value must have at least 3 rows (minimum for a polygon).

Value

The `name` string, invisibly.

Examples

```
my_blueprint <- data.frame(
  dx   = c(0, 1, 1, 0,   2, 3, 3, 2),
  dy   = c(0, 0, 1, 1,   0, 0, 1, 1),
  part = c(rep("SegA", 4), rep("SegB", 4))
)
register_colon_variant("my_colon", my_blueprint)
list_colon_variants()
```

Index

aes(), *4, 5*

colon_heatmap_grid, 2
colon_heatmap_grid(), *7*
colon_sf, 3
colon_sf(), *7*

fortify(), *4, 5*

geom_colon, 4
geom_colon(), *7*
geom_colon_label, 5
geom_colon_label(), *7*
ggcolonviz *(*ggcolonviz-package*)*, 2
ggcolonviz-package, 2
ggplot(), *4, 5*

list_colon_variants, 6

register_colon_variant, 7
register_colon_variant(), *6*