Package ‘OrgHeatmap’

March 31, 2026

Title Visualization Tool for Numerical Data on Human/Mouse Organs and Organelles

Version 0.3.4

Author Qirui Shen [aut, cre]

Maintainer Qirui Shen <`shenqr@i.smu.edu.cn`>

Description A tool for visualizing numerical data (e.g., gene expression, protein abundance) on predefined anatomical maps of human/mouse organs and subcellular organelles. It supports customization of color schemes, filtering by organ systems (for organisms) or organelle types, and generation of optional bar charts for quantitative comparison. The package integrates coordinate data for organs and organelles to plot anatomical/subcellular contours, mapping data values to specific structures for intuitive visualization of biological data distribution.The underlying method was described in the preprint by Zhou et al. (2022) <doi:10.1101/2022.09.07.506938>.

Depends R (>= 4.1.0), ggplot2 (>= 3.5.0)

Imports ggpolypath (>= 0.3.0), dplyr (>= 1.1.0), magrittr, rlang, stringr, patchwork, purrr, stringdist, sf, stats, utils, data.table, RColorBrewer, grDevices, viridis (>= 0.6.0)

Suggests knitr, svglite, rmarkdown, testthat (>= 3.0.0)

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

LazyData true

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2026-03-31 10:10:16 UTC

Contents

`human_bodycontour` *Human body contour coordinate data*

Description

Contains plotting coordinates for the human body contour, used for drawing the background outline.

Usage

```
human_bodycontour
```

Format

A data frame containing:

`V1` Integer. Original index from source data (reserved for compatibility).

`id` Character. Group identifier for contour segments (used to draw connected regions).

`x` Numeric. X-coordinate of contour points.

`y` Numeric. Y-coordinate of contour points.

`human_organ_coord` *Human organ coordinate data*

Description

Contains plotting coordinates (x, y) for each organ, used for drawing organ contours.

Usage

```
human_organ_coord
```

Format

A list where each element is a data frame containing:

x x-coordinate (numeric)

y y-coordinate (numeric)

id Group ID (for drawing polygons, integer)

`human_organ_systems` *Human Organ-system mapping data*

Description

Defines the physiological system each organ belongs to, used for filtering organs by system.

Usage

```
human_organ_systems
```

Format

A data frame with the following columns:

organ Organ name (character, standardized name)

system Belonging system (character, e.g., "circulatory" for circulatory system)

`mouse_bodycontour` *Mouse body contour coordinate data*

Description

Contains plotting coordinates for the mouse body contour, used for drawing the background outline in mouse visualization.

Usage

```
mouse_bodycontour
```

Format

A data frame containing:

`V1` Integer. Original index from source data (reserved for compatibility).

`id` Character. Group identifier for contour segments (used to draw connected regions).

`x` Numeric. X-coordinate of contour points.

`y` Numeric. Y-coordinate of contour points.

`mouse_organ_coord` *Mouse organ coordinate data*

Description

Contains plotting coordinates (x, y) for each mouse organ, used for drawing organ contours in mouse visualization.

Usage

```
mouse_organ_coord
```

Format

A list where each element is a data frame containing:

x x-coordinate (numeric)

y y-coordinate (numeric)

id Group ID (for drawing polygons, integer)

`mouse_organ_systems` *Mouse Organ-system mapping data*

Description

Defines the physiological system each mouse organ belongs to, used for filtering organs by system in mouse visualization.

Usage

```
mouse_organ_systems
```

Format

A data frame with the following columns:

organ Organ name (character, standardized name)

system Belonging system (character, e.g., "circulatory" for circulatory system)

`organelle_bodycontour` *Organelle Body Contour Data*

Description

A dataset containing the contour coordinates for organelle cell outline.

Usage

```
organelle_bodycontour
```

Format

A data frame with variables for cell contour coordinates

`organelle_organ_coord` *Organelle Organ Coordinate Data*

Description

A list containing coordinate data for various organelles.

Usage

```
organelle_organ_coord
```

Format

A named list where each element is a data frame with organelle coordinates

---

`OrgHeatmap` *Visualization Tool for Human, Mouse and Organelle Data*

Description

This tool visualizes numerical data (such as gene expression) on a human, mouse or organelle map. It supports custom color schemes, organ system filtering, and bar charts for quantitative comparison.

Usage

```
OrgHeatmap(
  data,
  species = c("human", "mouse", "organelle"),
  system = NULL,
  valid_organs = NULL,
  sort_by_value = TRUE,
  title = NULL,
  showall = FALSE,
  outline = TRUE,
  palette = "YlOrRd",
  color_high = NULL,
  color_low = NULL,
  color_mid = NULL,
  reverse_palette = FALSE,
  fillcolor_outline = NULL,
  fillcolor_organ = "plasma",
  fillcolor_other = "#D3D3D3",
  organbar = TRUE,
  organbar_title = NULL,
  organbar_digit = 4,
  organbar_color = NULL,
  organbar_text_color = "black",
  organbar_low = NULL,
  organbar_high = NULL,
  direction = 1,
  save_clean_data = FALSE,
  save_plot = FALSE,
  clean_data_path = file.path(getwd(), "clean_data.rds"),
  plot_path = file.path(getwd(), "organ_plot.png"),
  plot_width = 10,
  plot_height = 8,
  plot_dpi = 300,
  plot_device = "png",
  organ_system_map = NULL,
  organ_name_mapping = NULL,
  aggregate_method = "mean",
  organ_col = "organ",
  value_col = "value"
)
```

Arguments

`data`	Data frame with at least two columns: organ name and corresponding value
`species`	Character, species to visualize, one of `"human"` (default), `"mouse"`, or `"organelle"`. Determines the default body/cell contour, organ coordinates, and organ-system mapping.

`system`	Optional character vector specifying organ system to display(not applicable for organelles)
`valid_organs`	Optional character vector of valid organ names for filtering
`sort_by_value`	Logical, default TRUE, sorts by value descending
`title`	Optional character vector for plot title
`showall`	Logical, default FALSE. If TRUE, shows all organ outlines (grey) with light grey fill (#EFEFEF) for non-target organs (to provide anatomical context).
`outline`	Logical, default TRUE, draws human/mouse/cell outline
`palette`	Character, name of RColorBrewer palette (e.g., "YlOrRd", "PuBuGn") for unified color scheme (applies to both organ heatmap and bar chart if no custom colors are specified). Ignored if `color_low/color_high` (for heatmap) or `organbar_low/organbar_high` (for bar chart) are specified. Default: "YlOrRd" (suitable for highlighting high values).
`color_high`	Character, custom color for the **maximum value** of the organ heatmap (and bar chart if `organbar_high` is not specified). Overrides `palette` but is overridden by `organbar_high` (highest priority). Default: NULL.
`color_low`	Character, custom color for the **minimum value** of the organ heatmap (and bar chart if `organbar_low` is not specified). Overrides `palette` but is overridden by `organbar_low` (highest priority). Default: NULL.
`color_mid`	Character, optional color for the **middle value** of the organ heatmap (for 3-color gradients). Default: NULL.
`reverse_palette`	Logical, whether to reverse the color order of `palette`. Default: FALSE (low=light, high=dark).
`fillcolor_outline`	Character, default "#F5D5B8" for human/mouse, "#F0F8FF" for organelle, fill color for outline
`fillcolor_organ`	Character, fallback color scheme for organs (supports viridis options: "viridis", "plasma", "magma", etc.). Only used if no `palette`, `color_low/color_high`, or `organbar_low/organbar_high` are specified. Default: "plasma".
`fillcolor_other`	Character, default "#D3D3D3", fill color for non-target organelles
`organbar`	Logical, default TRUE, shows value bar chart
`organbar_title`	Optional character, title for bar chart legend
`organbar_digit`	Integer, default 4, digits for bar values
`organbar_color`	Optional character, solid color for bars
`organbar_text_color`	Character. The text color for the organ bar chart values. Default is "black".
`organbar_low`	Character, low end of gradient for **bar chart** (and organ heatmap if `color_low` is not specified). Highest priority for color configuration (overrides `color_low` and `palette`). Default: NULL.

`organbar_high`	Character, high end of gradient for **bar chart** (and organ heatmap if `color_high` is not specified). Highest priority for color configuration (overrides `color_high` and `palette`). Default: NULL.
`direction`	Integer, default 1. Direction of color gradient: 1 = normal (low value → light color, high value → dark color); -1 = reversed (low value → dark color, high value → light color).
`save_clean_data`	Logical, default FALSE, saves cleaned data
`save_plot`	Logical, default FALSE, whether to save the plot
`clean_data_path`	Character, default `file.path(getwd(), "clean_data.rds")`, path for cleaned data (default to current working directory)
`plot_path`	Character, default `file.path(getwd(), "organ_plot.png")`, path for saving the plot (default to current working directory)
`plot_width`	Numeric, default 10, plot width in inches
`plot_height`	Numeric, default 8, plot height in inches
`plot_dpi`	Numeric, default 300, plot resolution
`plot_device`	Character, default "png", plot format (e.g., "png", "pdf")
`organ_system_map`	Data frame, CSV path, or NULL (default). If NULL, uses species-specific defaults: `human_organ_systems` for humans, `mouse_organ_systems` for mice, or `organelle_systems` for organelles.
`organ_name_mapping`	Optional: Either a named vector (non-standard → standard names, e.g., c("adrenal" = "adrenal_gland")), a data frame (must contain columns specified by `original_col` and `standard_col`), or a CSV path (same column requirement). Internally processed by `create_organ_mapping()` for standardization (lowercase, underscores for spaces).
`aggregate_method`	Character, aggregation method for duplicate organs (one of "mean" `default`, "sum", "count").
`organ_col`	Character, default "organ", column name for organs
`value_col`	Character, default "value", column name for values

Details

The function uses `get_component_colors()` (an internal helper function) to generate unified color schemes:

1. If `organbar_low` and `organbar_high` are specified by the user, they will be used directly (highest priority);
2. If not, colors are generated from the `palette` (RColorBrewer) with optional reversal (`reverse_palette`);
3. Custom middle color (`color_mid`) is supported for 3-color gradients (applied to both heatmap and bar chart). For valid `palette` names, see `RColorBrewer::brewer.pal.info`.

Organ Mapping Logic:

- For `organ_name_mapping`: Accepts a named vector, data frame, or CSV path. Internal helper `create_organ_mapping()` standardizes names (lowercase, underscores for spaces).
- For `organ_system_map`: Custom tables (data frame/CSV) are processed to align with built-in `human_organ_systems` (for humans), `mouse_organ_systems` (for mice), or `organelle_systems` (for organelles) format via `create_organ_mapping()`.

Value

List containing:

`plot`	ggplot2 object
`clean_data`	Cleaned data frame
`system_used`	Organ system used
`mapped_organs`	Standardized organ names
`missing_organs`	Organs without coordinates
`total_value`	Sum of all values

Examples

```
# Load the package
library(OrgHeatmap)

# Note: Example datasets (example_Data1, example_Data2, example_Data3, expr_data)
# are included in the package's 'extdata' directory.

## Load example data files from extdata (contains example_Data1, example_Data2, example_Data3)
data_path <- system.file("extdata", "exampledata.Rdata", package = "OrgHeatmap")
load(data_path)

# 1.Plot all organs and save results using internal saving function
result_all <- OrgHeatmap(
  example_Data3,
  organbar = TRUE,
  save_plot = TRUE,  # Enable plot saving
  plot_path = file.path(tempdir(), "all_system.png"),
  plot_width = 10,
  plot_height = 8,
  save_clean_data = TRUE,  # Enable cleaned data saving
  clean_data_path = file.path(tempdir(), "all_system_clean_data.rds")
)
print(result_all$plot)  # Print the plot to the console

# 2. Plot circulatory system organs and save results
result_circulatory <- OrgHeatmap(
  example_Data3,
  system = "circulatory",
 organbar = TRUE,
  save_plot = TRUE,
 plot_path = file.path(tempdir(), "circulatory_system.png"),
```

```
 plot_width = 10,
  plot_height = 8,
  plot_device = "png",  # Specify plot format
  save_clean_data = TRUE,
  clean_data_path = file.path(tempdir(), "circulatory_clean_data.rds")
)
print(result_circulatory$plot)  # Print the plot to the console

# 3. Quick color configuration with palette
# Core logic: Trigger internal color_config generation via palette parameters,
# ensuring organ and bar chart colors are synchronized
result_palette <- OrgHeatmap(
example_Data3,
system = "respiratory",  # Respiratory system
palette = "PuBuGn",      # Use RColorBrewer's blue-purple-green palette
reverse_palette = TRUE,  # Reverse palette (low value = dark green, high value = purple)
color_mid = "#87CEEB",   # Custom middle color (sky blue)
title = "Respiratory System (Palette: PuBuGn)",
organbar_title = "Mean Value",
organbar_digit = 2,
showall = TRUE,
save_plot = TRUE,
plot_path = file.path(tempdir(),"respiratory_palette.png")
  # To use solid color for bars, add parameter: organbar_color = "skyblue"
  # (overrides gradient and synchronizes with organ colors)
)
print(result_palette$plot)


 ## Load the example dataset expr_data from extdata
 expr_data_path <- system.file("extdata", "expr_data.rds", package = "OrgHeatmap")
 expr_data <- readRDS(expr_data_path)

# Custom organ mapping (standardize original organ names)
custom_mapping <- c(
  "adrenal" = "adrenal_gland",
  "lymph node" = "lymph_node",
  "soft tissue" = "muscle",
  "peritoneal" = "peritoneum"
)

# Add prostate cancer-specific organs to system mapping
prostate_organ_systems <- rbind(
  human_organ_systems,  # Package's built-in organ system mapping
  data.frame(
    organ = c("prostate", "bone", "lymph_node", "adrenal_gland"),
    system = c("reproductive", "musculoskeletal", "lymphatic", "endocrine"),
    stringsAsFactors = FALSE
  )
)

# Generate TP53 expression heatmap and save results
tp53_plot <- OrgHeatmap(
```

```
  data = expr_data,
  value_col = ”expression”,  # Specify value column name as ”expression”
  organ_system_map = prostate_organ_systems,  # Use custom organ system mapping
  organ_name_mapping = custom_mapping,  # Apply organ name mapping
  title = ”TP53 Expression in Metastatic Prostate Cancer”,
  organbar_title = ”Mean Expression(log2) of TP53”,
  aggregate_method = ”mean”,  # Calculate mean by organ
  showall = TRUE,  # Show all organ outlines
  fillcolor_other = ”#DCDCDC”,  # Fill color for non-target organs
  organbar_digit = 2,  # Keep 2 decimal places for bar values
  direction = -1,  # Reverse color gradient (darker = higher expression)
  save_plot = TRUE,  # Save the plot
  plot_path = file.path(tempdir(), ”tp53_expression_metastatic_prostate.png”),
  plot_width = 14,
  plot_height = 10,
  plot_dpi = 300,
  save_clean_data = TRUE,  # Save cleaned data
  clean_data_path = file.path(tempdir(), ”tp53_clean_data.rds”)
)

# Print the plot
print(tp53_plot$plot)

# 4.Plot mouse digestive system
# Load mouse example data (included in the package)
mouse_data_path <- system.file(”extdata”, ”exampledata.Rdata”, package = ”OrgHeatmap”)
load(mouse_data_path)

# Generate plot for mouse digestive system
mouse_digestive_plot <- OrgHeatmap(
data = example_Data1,
species = ”mouse”,  # Specify mouse species
system = ”digestive”,
organbar = TRUE,
palette = ”PuBu”,
save_plot = TRUE,
plot_path = file.path(tempdir(), ”mouse_digestive_plot.png”),
save_clean_data = TRUE,
clean_data_path = file.path(tempdir(), ”mouse_digestive_data.rds”)
)
print(mouse_digestive_plot$plot)


 # 5. Plot organelles
organelle_data <- data.frame(
 organ = c(”mitochondrion”, ”nucleus”, ”endoplasmic_reticulum”,”cell_membrane”),
 value = c(15.2, 8.7, 6.3,6.8)
)

organelle_plot <- OrgHeatmap(
 data = organelle_data,
 species = ”organelle”,
title = ”Organelle Expression Visualization”,
```

```
organbar_title = "Expression Level",
save_plot = TRUE,
plot_path = file.path(tempdir(), "organelle_expression_plot.png")
)
```

Index