

# Package ‘escheR’

January 27, 2025

**Title** Unified multi-dimensional visualizations with Gestalt principles

**Version** 1.6.0

**Description** The creation of effective visualizations is a fundamental component of data analysis. In biomedical research, new challenges are emerging to visualize multi-dimensional data in a 2D space, but current data visualization tools have limited capabilities. To address this problem, we leverage Gestalt principles to improve the design and interpretability of multi-dimensional data in 2D data visualizations, layering aesthetics to display multiple variables. The proposed visualization can be applied to spatially-resolved transcriptomics data, but also broadly to data visualized in 2D space, such as embedding visualizations. We provide this open source R package escheR, which is built off of the state-of-the-art ggplot2 visualization framework and can be seamlessly integrated into genomics toolboxes and workflows.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**biocViews** Spatial, SingleCell, Transcriptomics, Visualization,  
Software

**Depends** ggplot2, R (>= 4.3)

**Imports** SpatialExperiment (>= 1.6.1), SingleCellExperiment, rlang,  
SummarizedExperiment

**BugReports** <https://github.com/boyigu01/escheR/issues>

**URL** <https://github.com/boyigu01/escheR>

**Suggests** STexampleData, BumpyMatrix, knitr, rmarkdown, BiocStyle,  
ggpubr, scran, scatter, scuttle, Seurat, hexbin

**VignetteBuilder** knitr

**git\_url** <https://git.bioconductor.org/packages/escheR>

**git\_branch** RELEASE\_3\_20

**git\_last\_commit** 70999bb

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.20

**Date/Publication** 2025-01-27

**Author** Boyi Guo [aut, cre] (<<https://orcid.org/0000-0003-2950-2349>>),  
 Stephanie C. Hicks [aut] (<<https://orcid.org/0000-0002-7858-0231>>),  
 Erik D. Nelson [ctb] (<<https://orcid.org/0000-0001-8477-0982>>)

**Maintainer** Boyi Guo <boyi.guo.work@gmail.com>

## Contents

.contain_reserved_col_name	2
add_fill	2
add_ground	3
add_symbol	4
make_escheR	5
<b>Index</b>	<b>7</b>

---

.contain_reserved_col_name	<i>Check if rowData(spe) contains reserved name</i>
----------------------------	---

---

### Description

Internal Funciton

### Usage

```
.contain_reserved_col_name(col_name)
```

### Arguments

col\_name      the colnames

### Value

TRUE when col\_name contains reserved names, FALSE

---

add_fill	<i>Adding fill to highlight the figure in the spatial map</i>
----------	---

---

### Description

Adding fill to highlight the figure in the spatial map

### Usage

```
add_fill(p, var, point_size = 2, ...)
```

```
add_fill_bin(p, var, bins = 30, point_size = 2.8, fun = sum, ...)
```

**Arguments**

p	a spatial map created by <code>make_escheR()</code> , with or without other layers of aesthetics.
var	A character(1) with the name of the <code>colData(spe)</code> column that has the values to be used as the background.
point_size	A numeric(1) specifying the size of the spot in the ggplot. Defaults to 2.
...	Reserved for future arguments.
bins	numeric vector giving number of bins in both vertical and horizontal directions. Set to 30 by default.
fun	function for summary. See more detail in <a href="#">stat_summary_hex</a>

**Value**

an ggplot object.

**Examples**

```
library(STexampleData)

spe <- Visium_humanDLPFC()

make_escheR(spe) |>
  add_fill(var = "ground_truth")
```

---

add\_ground

*Adding border to highlight the ground in the spatial map*

---

**Description**

Adding border to highlight the ground in the spatial map

**Usage**

```
add_ground(p, var, stroke = 0.5, point_size = 2, ...)
```

```
add_ground_bin(p, var, bins = 30, stroke = 1, point_size = 3, ...)
```

**Arguments**

p	a spatial map created by <code>make_escheR()</code> , with or without other layers of aesthetics.
var	A character(1) with the name of the <code>colData(spe)</code> column that has the values to be used as the background.
stroke	A numeric(1) specifying the thickness of the border.
point_size	A numeric(1) specifying the size of the spot in the ggplot. Defaults to 2.
...	Reserved for future arguments.
bins	numeric vector giving number of bins in both vertical and horizontal directions. Set to 30 by default.

**Value**

an ggplot object.

**Examples**

```
library(STexampleData)

spe <- Visium_humanDLPFC()

make_escheR(spe) |>
  add_ground(var = "ground_truth")
```

---

 add\_symbol

*Adding symbols to each spot in the spatial map*


---

**Description**

Adding symbols to each spot in the spatial map

**Usage**

```
add_symbol(p, var, size = 1, ...)
```

**Arguments**

p	a spatial map created by <code>make_escheR()</code> , with or without other layers of aesthetics.
var	A <code>character(1)</code> with the name of the <code>colData(spe)</code> column that has the values to be used as the background.
size	A <code>numeric(1)</code> specifying the size of the symbols in the ggplot. Defaults to 1.
...	Reserved for future arguments.

**Value**

an ggplot object.

**Examples**

```
library(STexampleData)

spe <- Visium_humanDLPFC()

# Convert a continuous variable to categorical
spe$in_tissue <- factor(spe$in_tissue)

make_escheR(spe) |>
  add_ground(var = "ground_truth") |>
  add_symbol(var = "in_tissue", size = 0.5)
```

---

`make_escheR`*Create a new spatial map for spatial transcriptomics data*

---

## Description

`make_escheR()` is a generic function to initialize a `ggplot` object that contains a spatial map. Because the `ggplot` object saves the input spatial transcriptomics data, the transcriptomics data will be used in the following layering process to add more aesthetic components in the plot following the grammar of graphics and `ggplot2` syntax.

## Usage

```
make_escheR(object, spot_size = 2, ...)  
  
## S3 method for class 'SingleCellExperiment'  
make_escheR(object, spot_size = 2, dimred = "PCA", ...)  
  
## S3 method for class 'SpatialExperiment'  
make_escheR(object, spot_size = 2, dimred = NULL, y_reverse = TRUE, ...)  
  
## S3 method for class 'data.frame'  
make_escheR(object, spot_size = 2, .x, .y, ...)
```

## Arguments

<code>object</code>	a data object that contains the spatial transcriptomics data. Currently only working for spatial transcriptomics data as <code>SpatialExperiment</code> objects.
<code>spot_size</code>	A <code>numeric(1)</code> specifying the size of the spot in the <code>ggplot</code> . Defaults to 2.
<code>...</code>	Reserved for future arguments.
<code>dimred</code>	String or integer scalar specifying the existing dimensionality reduction results to use.
<code>y_reverse</code>	(logical) Whether to reverse y coordinates, which is often required for 10x Genomics Visium data. Default = <code>TRUE</code> .
<code>.x</code>	the X-coordinate
<code>.y</code>	the Y-coordinate

## Value

an `ggplot` object that contains the spatial transcriptomics data.

## References

Guo B, Huuki-Myers LA, Grant-Peters M, Collado-Torres L, Hicks SC (2023). `escheR`: Unified multi-dimensional visualizations with Gestalt principles. *Bioinformatics Advances*, Volume 3, Issue 1, vbad179, doi:10.1093/bioadv/vbad179

**Examples**

```
library(STexampleData)
# SpatialExperiment Object
spe <- Visium_humanDLPFC()
make_escheR(spe)

# SingleCellExperiment Object
sce <- SingleCellExperiment(counts(spe))
reducedDims(sce) <- list(
  # Example embedding
  EG = matrix(seq.int(1, ncol(spe)*2), ncol = 2)
)
make_escheR(sce, dimred = "EG")

# data.frame Object
x <- spatialCoords(spe)[,1]
y <- spatialCoords(spe)[,2]
df <- colData(spe) |> data.frame()
make_escheR(object = df, .x = x, .y = y)
```

# Index

`.contain_reserved_col_name`, [2](#)

`add_fill`, [2](#)

`add_fill_bin (add_fill)`, [2](#)

`add_ground`, [3](#)

`add_ground_bin (add_ground)`, [3](#)

`add_symbol`, [4](#)

`make_escheR`, [5](#)

`stat_summary_hex`, [3](#)