

# Package: scOverlay (via r-universe)

June 22, 2026

**Type** Package

**Title** Multilayer plots for single-cell data

**Version** 0.99.0

**Description** scOverlay provides publication-grade plotting helpers to display a foreground feature (gene expression or metadata) on top of a contextual background (clusters, cell types, samples, or a solid colour) in the same reduced dimension panel, each with independent colour scales (via ggnewscale). It integrates with SingleCellExperiment / SummarizedExperiment, supports flexible subsetting of foreground cells, multiple assays (e.g. counts, logcounts), and grid arrangements by user-defined grouping columns.

**License** Artistic-2.0

**Encoding** UTF-8

**Depends** R (>= 4.5.0)

**Imports** ggplot2, ggnewscale, SummarizedExperiment, SingleCellExperiment, viridisLite, scales, RColorBrewer, patchwork, ggrastr, rlang

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0), BiocStyle

**VignetteBuilder** knitr

**Roxygen** list(markdown = TRUE)

**URL** <https://github.com/bernatgel/scOverlay>

**BugReports** <https://github.com/bernatgel/scOverlay/issues>

**biocViews** Software, Visualization, SingleCell, Transcriptomics, GeneExpression, DataRepresentation

**Config/testthat/edition** 3

**Config/roxygen2/version** 8.0.0

**Config/pak/sysreqs** libcairo2-dev libfontconfig1-dev libfreetype6-dev libfribidi-dev libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libwebp-dev zlib1g-dev

**Repository** <https://biocstaging.r-universe.dev>

**Date/Publication** 2026-06-02 16:25:12 UTC

**RemoteUrl** <https://github.com/BiocStaging/scOverlay>

**RemoteRef** HEAD

**RemoteSha** a1e68884d4f118b0d2a8fc2588aceb464fba3935

## Contents

getPalette . . . . .	2
listPalettes . . . . .	4
plotGeneOverlay . . . . .	5
plotOverlay . . . . .	6
plotOverlayPerGroup . . . . .	11
plotOverlayPerNestedGroup . . . . .	15
sce_mpnst_example . . . . .	20
sce_overlay_example . . . . .	21

**Index** [22](#)

---

getPalette	<i>Resolve a scOverlay palette</i>
------------	------------------------------------

---

## Description

getPalette converts a palette specification into a vector of colours that can be used by scOverlay or directly in a **ggplot2** scale.

## Usage

```
getPalette(
  palette = NULL,
  type = c("continuous", "categorical"),
  n = NULL,
  values = NULL
)
```

## Arguments

palette	Palette specification. It can be NULL, a single palette name, a vector of colours or a palette function. A palette function should take the number of requested colours as its first argument and return a colour vector. Defaults to NULL.
type	Character value indicating the type of palette to return. Accepted values are "continuous" and "categorical". Defaults to "continuous".
n	Number of colours to return. If NULL, it is inferred from values when available, otherwise a default of 256 is used. Non-integer values of n are truncated with <code>as.integer()</code> . Defaults to NULL.

values            Optional character vector with the expected categorical values. When supplied and the resolved palette is named, names are used to map colours to values. Defaults to NULL.

## Details

Palette specifications can be NULL, the name of one of the palettes bundled with `scOverlay`, a vector of colours, or a palette function. Palette names are first looked up in `scOverlay`'s internal palette registry for the requested type, then in **viridisLite**, and then among ColorBrewer palettes through **scales**.

When `palette = NULL`, `getPalette` uses the default `scOverlay` palette for the requested type: `"scOverlay"` for categorical palettes and `"gray_red"` for continuous palettes.

Continuous palettes are represented as anchor colours and interpolated internally to the requested number of colours. This means that a user supplied vector such as `c("gray90", "red", "darkred")` can be used as a continuous gradient.

Categorical palettes can be named. When `values` is supplied and all values are present in the palette names, colours are matched by name. When no values are present in the palette names, colours are assigned by position. If only some values match, `getPalette()` raises an error to avoid silent colour remapping. Built-in categorical palettes intentionally keep integer-like names such as `"1"`, `"2"` and `"3"` because these are useful for cluster labels. Use `unnamed()` to force positional matching for a named palette, or `setNames()` to assign or replace names.

If a categorical palette has fewer colours than requested, additional colours are generated by interpolation.

## Value

A character vector of colours. For categorical palettes, the returned vector is named with values when `values` is supplied.

## See Also

[listPalettes](#), [plotOverlay](#).

## Examples

```
getPalette("gray_red", type = "continuous", n = 5)
```

```
getPalette(
  palette = c("#E6E6E6", "#EF6F6C", "#67000D"),
  type = "continuous",
  n = 5
)
```

```
getPalette(
  palette = "scOverlay",
  type = "categorical",
  n = 3,
  values = c("A", "B", "C")
)
```

```

getPalette(
  palette = c(A = "red", B = "blue"),
  type = "categorical",
  values = c("B", "A")
)

pal <- getPalette("scOverlay", type = "categorical", n = 4)
setNames(unnamed(pal), c("Nerve", "PNF", "ANF", "MPNST"))

```

---

listPalettes	<i>List available scOverlay palettes</i>
--------------	--

---

### Description

listPalettes lists the palettes bundled with scOverlay or returns a compact graphical preview of them.

### Usage

```

listPalettes(
  type = c("all", "categorical", "continuous"),
  plot = FALSE,
  n = 24,
  show_names = TRUE
)

```

### Arguments

type	Character value indicating which palettes to list. Accepted values are "all", "categorical" and "continuous". Defaults to "all".
plot	Logical value. If FALSE, return a data frame with palette metadata. If TRUE, return a <b>ggplot2</b> swatch preview. Defaults to FALSE.
n	Number of colours used to display continuous palettes, and the maximum number of colours shown for categorical palettes when plot = TRUE. Non-integer values of n are truncated with <code>as.integer()</code> . Defaults to 24.
show_names	Logical value. If TRUE, categorical palette previews show stored colour names when available, or position labels otherwise. Continuous palette previews do not show per-colour labels. Defaults to TRUE.

### Details

The function can return either a small metadata table or a **ggplot2** swatch plot. Categorical palettes are shown as their stored colours. Continuous palettes are stored internally as anchor colours and are interpolated to n colours when plot = TRUE.

When `plot = TRUE` and `show_names = TRUE`, categorical palette swatches show their stored names when names are present. Built-in categorical palettes intentionally keep integer-like names because these are useful for cluster labels.

The `is_default` column marks the palettes used by `getPalette` when `palette = NULL`: "scOverlay" for categorical palettes and "gray\_red" for continuous palettes.

### Value

If `plot = FALSE`, a `data.frame` with palette names, types, number of stored colours and a logical column indicating the default palettes. If `plot = TRUE`, a **ggplot2** object with a swatch preview.

### See Also

[getPalette](#), [plotOverlay](#).

### Examples

```
listPalettes()

listPalettes(type = "continuous")

p <- listPalettes(plot = TRUE, n = 16, show_names = TRUE)
p
```

---

plotGeneOverlay	<i>Plot overlays for several genes</i>
-----------------	--

---

### Description

`plotGeneOverlay` creates one [plotOverlay](#) plot for each gene in `genes` and returns them as a named list.

### Usage

```
plotGeneOverlay(sce, genes, ...)
```

### Arguments

<code>sce</code>	A <a href="#">SingleCellExperiment</a> object containing the reduced dimensions and gene expression values to plot.
<code>genes</code>	Character vector with the genes to plot. Genes are looked up in <code>rownames(sce)</code> . This argument has no default and must be provided by the user.
<code>...</code>	Additional arguments passed to <a href="#">plotOverlay</a> , such as <code>reduced_dim</code> , <code>background</code> , <code>bg_type</code> , <code>bg_assay</code> , <code>fg_assay</code> , <code>fg_order</code> , <code>palettes</code> or <code>rasterization</code> options.

## Details

This is a convenience function to quickly inspect the expression pattern of several genes on the same embedding. All additional plotting arguments are passed to [plotOverlay](#).

Genes not present in `rownames(sce)` are skipped with a warning. If none of the requested genes are found, the function stops with an error.

## Value

A named list of **ggplot2** objects, one for each gene found in `rownames(sce)`.

## See Also

[plotOverlay](#), [getPalette](#) and [listPalettes](#).

## Examples

```
data("sce_overlay_example")

plots <- plotGeneOverlay(
  sce = sce_overlay_example,
  genes = c("SOX10", "S100B"),
  reduced_dim = "TSNE",
  background = "cluster",
  fg_order = "ascending"
)

plots$SOX10
```

---

plotOverlay

*Plot layered information on a single-cell embedding*

---

## Description

`plotOverlay` creates a layered plot of a reduced dimension representation stored in a [SingleCellExperiment](#) object. It first creates a background layer, dims this background with a semi-transparent white overlay, and finally draws a foreground variable on top using an independent colour scale. When a layer is set to "none", the corresponding point layer is kept in the plot structure without drawing visible points.

## Usage

```
plotOverlay(
  sce,
  foreground,
  background = "solid",
  reduced_dim = "TSNE",
  fg_subset_cells = NULL,
```

```

bg_type = c("auto", "continuous", "categorical"),
fg_type = c("auto", "continuous", "categorical"),
bg_assay = "logcounts",
fg_assay = "logcounts",
fg_order = c("input", "random", "ascending", "descending"),
bg_point_size = 4,
fg_point_size = 1,
bg_palette = NULL,
fg_palette = NULL,
bg_raster = FALSE,
fg_raster = FALSE,
raster_dpi = 300,
bg_dimming = 0.9,
fg_limits = NULL,
bg_limits = NULL,
bg_legend = TRUE,
fg_legend = TRUE,
bg_legend_title = NULL,
fg_legend_title = NULL,
title = NULL,
bg_values = NULL,
fg_values = NULL
)

```

## Arguments

sce	A <a href="#">SingleCellExperiment</a> object containing the reduced dimensions and the data to plot. The object must contain the reduced dimension specified in <code>reduced_dim</code> . If background or foreground is a feature, the object must also contain the corresponding assay specified in <code>bg_assay</code> or <code>fg_assay</code> .
foreground	A single character value specifying the variable to draw in the foreground layer. It can be "solid" to draw selected foreground cells with a fixed colour, "none" to draw no visible foreground points, a feature in <code>rownames(sce)</code> , in which case values are taken from <code>fg_assay</code> , or a column in <code>colData(sce)</code> . This argument has no default and must be provided by the user.
background	A single character value specifying the background layer. It can be "solid" to draw all background cells with a fixed colour, "none" to draw no visible background points, a feature in <code>rownames(sce)</code> , or a column in <code>colData(sce)</code> . Defaults to "solid".
reduced_dim	A single character value with the name of the reduced dimension to plot. It must be one of <code>reducedDimNames(sce)</code> . Only the first two dimensions are used. Defaults to "TSNE".
fg_subset_cells	A cell subset defining which cells are shown in the foreground layer. It can be NULL, a logical vector of length <code>ncol(sce)</code> , a character vector with cell names, a function receiving <code>sce</code> and returning a logical vector, or an expression evaluated in <code>colData(sce)</code> . NULL selects all cells. NA values are treated as FALSE. If

	no cells match, an empty foreground layer is drawn and a warning is emitted. Defaults to NULL.
bg_type	A character value indicating how the background variable should be represented when background is not "solid" or "none". Accepted values are "auto", "continuous" and "categorical". With "auto", feature values are treated as continuous and colData variables are treated as continuous only when they are numeric. Defaults to "auto".
fg_type	A character value indicating how the foreground variable should be represented when foreground is not "solid" or "none". Accepted values are "auto", "continuous" and "categorical". With "auto", feature values are treated as continuous and colData variables are treated as continuous only when they are numeric. Defaults to "auto".
bg_assay	A single character value with the name of the assay to use when background is a feature in rownames(sce). It is ignored when background = "solid" or background = "none". Defaults to "logcounts".
fg_assay	A single character value with the name of the assay to use when foreground is a feature in rownames(sce). It is ignored when foreground = "solid" or foreground = "none". Defaults to "logcounts".
fg_order	A character value controlling the drawing order of foreground cells. Accepted values are "input", "random", "ascending" and "descending". "input" keeps the current order, "random" randomly shuffles foreground cells, "ascending" draws lower foreground values first and higher values on top, and "descending" does the opposite. "ascending" is useful for expression plots because cells with higher expression are drawn last. For reproducible random ordering, call set.seed() before plotOverlay(). Defaults to "input".
bg_point_size	Numeric value with the point size of the background layer. Defaults to 4.
fg_point_size	Numeric value with the point size of the foreground layer. Defaults to 1.
bg_palette	Palette specification for the background layer. It is ignored when background = "none". When background = "solid", NULL uses "gray70" and a single valid colour can be used to set the solid background colour. For variable backgrounds, it can be NULL, a palette name, a vector of colours or a palette function, and the value is passed to <code>getPalette</code> . Defaults to NULL, which uses "gray_red" for continuous backgrounds and "scOverlay" for categorical backgrounds.
fg_palette	Palette specification for the foreground layer. It is ignored when foreground = "none". When foreground = "solid", NULL uses "red" and a single valid colour can be used to set the solid foreground colour. For variable foregrounds, it can be NULL, a palette name, a vector of colours or a palette function, and the value is passed to <code>getPalette</code> . For continuous foregrounds, colour vectors are interpreted as gradient anchors and interpolated internally. Defaults to NULL, which uses "gray_red" for continuous foregrounds and "scOverlay" for categorical foregrounds.
bg_raster	Logical value. If TRUE, rasterize the background point layer using <code>ggrastr::geom_point_rast()</code> . Defaults to FALSE.
fg_raster	Logical value. If TRUE, rasterize the foreground point layer using <code>ggrastr::geom_point_rast()</code> . Defaults to FALSE.

raster_dpi	Positive numeric value with the resolution, in dots per inch, used for rasterized point layers. Defaults to 300.
bg_dimming	Numeric value between 0 and 1 controlling the opacity of the white dimming layer placed between background and foreground. A value of 0 does not dim the background and a value of 1 completely covers it with white. Defaults to 0.9.
fg_limits	Limits for the foreground colour scale when the foreground is continuous. It can be NULL, a strictly increasing numeric vector of length two, or "shared". In a single plot, "shared" resolves to the range of foreground values drawn in that plot. It is ignored with a warning when the foreground layer is not continuous. Defaults to NULL.
bg_limits	Limits for the background colour scale when the background is continuous. It can be NULL, a strictly increasing numeric vector of length two, or "shared". In a single plot, "shared" resolves to the range of background values in that plot. It is ignored with a warning when the background layer is not continuous. Defaults to NULL.
bg_legend	Logical value indicating whether to show the background legend. It is ignored when background = "solid" or background = "none". Defaults to TRUE.
fg_legend	Logical value indicating whether to show the foreground legend. It is ignored when foreground = "solid" or foreground = "none". Defaults to TRUE.
bg_legend_title	Character value with the title of the background legend. If NULL, background is used. Defaults to NULL.
fg_legend_title	Character value with the title of the foreground legend. If NULL, foreground is used. Defaults to NULL.
title	Character value with the plot title. If NULL, foreground is used, except when foreground = "none", where an empty title is used. Defaults to NULL.
bg_values	Categorical value set for the background scale. This is used by grouped plotting helpers to keep categorical colours consistent across panels. Not intended for users.
fg_values	Categorical value set for the foreground scale. This is used by grouped plotting helpers to keep categorical colours consistent across panels. Not intended for users.

## Details

This representation is useful to highlight gene expression, scores, annotations or selected groups of cells while keeping the complete embedding visible as context.

The function uses two visual layers. The background layer is used to show the global structure of the embedding, usually coloured by a cell annotation such as a cluster, sample or cell type. Either layer can also be set to "none" to keep the layer position without drawing visible points. The foreground layer is then drawn on top and can represent a fixed-colour selection, a feature from one of the assays, typically a gene expression value, or a variable stored in `colData(sce)`.

The foreground can be restricted to a subset of cells using `fg_subset_cells`. This only affects the foreground layer: cells not selected by `fg_subset_cells` are still shown in the background. This is

the main idea of the overlay plot and allows highlighting a particular group of cells without losing the context of the whole dataset.

The foreground colour scale is independent from the background colour scale. Palettes can be specified by name, as a vector of colours, or as a palette function. See [getPalette](#) and [listPalettes](#) for the available palettes and palette handling rules. For continuous palettes, colour vectors are used as anchors and interpolated internally.

For dense embeddings, the background and foreground point layers can be rasterized independently with `bg_raster` and `fg_raster`. Only the point layers are rasterized; axes, labels, legends and other plot elements remain vector graphics.

### Value

A **ggplot2** object with the background, dimming and foreground layers.

### See Also

[plotGeneOverlay](#), [plotOverlayPerGroup](#), [plotOverlayPerNestedGroup](#), [getPalette](#) and [listPalettes](#).

### Examples

```
data("sce_overlay_example")

# Overlay the expression of a gene on top of the cluster annotation.
p1 <- plotOverlay(
  sce = sce_overlay_example,
  reduced_dim = "TSNE",
  background = "cluster",
  foreground = "SOX10"
)
p1

# Use a solid background and draw higher expression values on top.
p2 <- plotOverlay(
  sce = sce_overlay_example,
  reduced_dim = "TSNE",
  background = "solid",
  foreground = "S100B",
  fg_order = "ascending"
)
p2

# Restrict the foreground to one sample type while keeping all cells in the
# background.
p3 <- plotOverlay(
  sce = sce_overlay_example,
  reduced_dim = "UMAP",
  background = "cluster",
  foreground = "SOX10",
  fg_subset_cells = quote(sample_type == "ANNUBP")
)
p3
```

```
# Use a custom continuous palette and rasterize the background point layer.
p4 <- plotOverlay(
  sce = sce_overlay_example,
  reduced_dim = "TSNE",
  background = "sample_type",
  foreground = "SOX10",
  fg_palette = c("#E6E6E6", "#EF6F6C", "#67000D"),
  bg_raster = TRUE
)
p4
```

---

plotOverlayPerGroup *Plot overlays split by a cell metadata group*

---

### Description

plotOverlayPerGroup creates one [plotOverlay](#) plot for each group defined by a column in `colData(sce)` and combines the plots into a single **patchwork** object.

### Usage

```
plotOverlayPerGroup(
  sce,
  group_col,
  foreground,
  background = "solid",
  reduced_dim = "TSNE",
  group_order = NULL,
  fg_subset_cells = NULL,
  bg_type = c("auto", "continuous", "categorical"),
  bg_assay = "logcounts",
  fg_assay = "logcounts",
  fg_type = c("auto", "continuous", "categorical"),
  fg_order = c("input", "random", "ascending", "descending"),
  bg_point_size = 2,
  fg_point_size = 0.5,
  bg_palette = NULL,
  fg_palette = NULL,
  bg_raster = FALSE,
  fg_raster = FALSE,
  raster_dpi = 300,
  bg_dimming = 0.6,
  fg_limits = NULL,
  bg_limits = NULL,
  bg_legend = TRUE,
  fg_legend = TRUE,
```

```

    bg_legend_title = NULL,
    fg_legend_title = NULL,
    shared_legend = FALSE
  )

```

### Arguments

sce	A <a href="#">SingleCellExperiment</a> object containing the reduced dimensions and the data to plot. The object must contain the reduced dimension specified in <code>reduced_dim</code> , the grouping column specified in <code>group_col</code> , and the data required by foreground.
group_col	A single character value with the name of the <code>colData(sce)</code> column used to split the plot into panels. Each value in this column defines one panel. This argument has no default and must be provided by the user.
foreground	A single character value specifying the variable to draw in the foreground layer. It can be "solid" to draw selected foreground cells with a fixed colour, "none" to draw no visible foreground points, a feature in <code>rownames(sce)</code> , in which case values are taken from <code>fg_assay</code> , or a column in <code>colData(sce)</code> . This argument has no default and must be provided by the user.
background	A single character value specifying the background layer passed to <a href="#">plotOverlay</a> . It can be "solid" to draw all background cells with a fixed colour, "none" to draw no visible background points, a feature in <code>rownames(sce)</code> , or a column in <code>colData(sce)</code> . Defaults to "solid".
reduced_dim	A single character value with the name of the reduced dimension to plot. It must be one of <code>reducedDimNames(sce)</code> . Only the first two dimensions are used. Defaults to "TSNE".
group_order	Optional character vector specifying the order in which groups should be plotted. If NULL, factor levels are used for factor grouping variables; otherwise, the order of appearance in <code>group_col</code> is used. Defaults to NULL.
fg_subset_cells	An optional additional cell subset applied to the foreground layer before splitting by group. It can be NULL, a logical vector of length <code>ncol(sce)</code> , a character vector with cell names, a function receiving <code>sce</code> and returning a logical vector, or an expression evaluated in <code>colData(sce)</code> . NULL selects all cells. NA values are treated as FALSE. Defaults to NULL. Groups with no matching foreground cells still produce panels with empty foreground layers.
bg_type	A character value indicating how the background variable should be represented when background is not "solid" or "none". Accepted values are "auto", "continuous" and "categorical". See <a href="#">plotOverlay</a> for details. Defaults to "auto".
bg_assay	A single character value with the name of the assay to use when background is a feature in <code>rownames(sce)</code> . It is ignored when <code>background = "solid"</code> or <code>background = "none"</code> . Defaults to "logcounts".
fg_assay	A single character value with the name of the assay to use when foreground is a feature in <code>rownames(sce)</code> . It is ignored when <code>foreground = "solid"</code> or <code>foreground = "none"</code> . Defaults to "logcounts".

fg_type	A character value indicating how the foreground variable should be represented when foreground is not "solid" or "none". Accepted values are "auto", "continuous" and "categorical". With "auto", feature values are treated as continuous and colData variables are treated as continuous only when they are numeric. Defaults to "auto".
fg_order	A character value controlling the drawing order of foreground cells within each panel. Accepted values are "input", "random", "ascending" and "descending". See <a href="#">plotOverlay</a> for details. Defaults to "input".
bg_point_size	Numeric value with the point size of the background layer in each panel. Defaults to 2.
fg_point_size	Numeric value with the point size of the foreground layer in each panel. Defaults to 0.5.
bg_palette	Palette specification for the background layer. It can be NULL, a palette name, a vector of colours or a palette function. When background = "solid", NULL uses "gray70" and a single valid colour can be used to set the solid background colour. It is ignored when background = "none". For variable backgrounds, the value is passed to <a href="#">getPalette</a> . Defaults to NULL, which uses "gray_red" for continuous backgrounds and "scOverlay" for categorical backgrounds.
fg_palette	Palette specification for the foreground layer. When foreground = "solid", NULL uses "red" and a single valid colour can be used to set the solid foreground colour. It is ignored when foreground = "none". For variable foregrounds, it can be NULL, a palette name, a vector of colours or a palette function. The value is passed to <a href="#">getPalette</a> . For continuous foregrounds, colour vectors are interpreted as gradient anchors and interpolated internally. Defaults to NULL, which uses "gray_red" for continuous foregrounds and "scOverlay" for categorical foregrounds.
bg_raster	Logical value. If TRUE, rasterize the background point layer in each panel using <code>ggrastr::geom_point_rast()</code> . Defaults to FALSE.
fg_raster	Logical value. If TRUE, rasterize the foreground point layer in each panel using <code>ggrastr::geom_point_rast()</code> . Defaults to FALSE.
raster_dpi	Positive numeric value with the resolution, in dots per inch, used for rasterized point layers. Defaults to 300.
bg_dimming	Numeric value between 0 and 1 controlling the opacity of the white dimming layer placed between background and foreground in each panel. A value of 0 does not dim the background and a value of 1 completely covers it with white. Defaults to 0.6.
fg_limits	Limits for the foreground colour scale when the foreground is continuous. It can be NULL, a strictly increasing numeric vector of length two, or "shared". With "shared", one range is computed after the global <code>fg_subset_cells</code> filter and reused in every panel. Defaults to NULL.
bg_limits	Limits for the background colour scale when the background is continuous. It can be NULL, a strictly increasing numeric vector of length two, or "shared". With "shared", one range is computed over all cells and reused in every panel. Defaults to NULL.
bg_legend	Logical value indicating whether to show the background legend in each panel. It is ignored when background = "solid". Defaults to TRUE.

<code>fg_legend</code>	Logical value indicating whether to show the foreground legend in each panel. It is ignored when <code>foreground = "solid"</code> . Defaults to <code>TRUE</code> .
<code>bg_legend_title</code>	Character value with the title of the background legend. If <code>NULL</code> , background is used. Defaults to <code>NULL</code> .
<code>fg_legend_title</code>	Character value with the title of the foreground legend. If <code>NULL</code> , foreground is used. Defaults to <code>NULL</code> .
<code>shared_legend</code>	Logical value indicating whether compatible legends should be collected into a shared patchwork legend. Defaults to <code>FALSE</code> .

### Details

This function is useful to compare the same foreground variable across samples, sample types, clusters or any other cell-level annotation. Each panel keeps the full embedding as background, while the foreground layer is restricted to the cells belonging to the corresponding group.

The grouping variable is specified with `group_col`. For each value in this column, `plotOverlayPerGroup` calls `plotOverlay` using the same background, foreground, palette and plotting options. The foreground subset for each panel is defined by the group membership and, optionally, by the additional filter supplied with `fg_subset_cells`.

As in `plotOverlay`, subsetting only affects the foreground layer. Unless `background = "none"`, the background layer still shows all cells in every panel, providing a common reference embedding across groups. This makes it possible to compare where cells from different samples or sample types are located in the same reduced dimension space.

With `fg_limits = "shared"`, one foreground range is computed after the global `fg_subset_cells` filter and before splitting into groups. With `bg_limits = "shared"`, one background range is computed across all cells. Categorical foreground and background palettes are based on global categorical values, so colours remain consistent across panels. Named categorical palettes are recommended when strict category-colour control is needed.

If `shared_legend = TRUE`, compatible legends are collected with `patchwork`. The legend can be repositioned afterwards with patchwork syntax, for example `p & ggplot2::theme(legend.position = "bottom")`.

If `group_order` is provided, panels are drawn following that order. If it is `NULL`, factor levels are used when `group_col` is a factor; otherwise, groups are shown in their order of appearance in the object.

### Value

A `patchwork` object combining one `plotOverlay` plot per group. If no valid groups produce a panel, the function returns `NULL` with a warning.

### See Also

`plotOverlay`, `plotOverlayPerNestedGroup`, `plotGeneOverlay`, `getPalette` and `listPalettes`.

## Examples

```
data("sce_overlay_example")

# Plot SOX10 expression separately for each sample type.
p1 <- plotOverlayPerGroup(
  sce = sce_overlay_example,
  foreground = "SOX10",
  group_col = "sample_type",
  reduced_dim = "TSNE",
  background = "solid",
  fg_order = "ascending"
)
p1

# Use the same grouping but colour the background by cluster.
p2 <- plotOverlayPerGroup(
  sce = sce_overlay_example,
  foreground = "S100B",
  group_col = "sample_type",
  reduced_dim = "TSNE",
  background = "cluster",
  fg_order = "ascending"
)
p2

# Specify an explicit group order.
p3 <- plotOverlayPerGroup(
  sce = sce_overlay_example,
  foreground = "SOX10",
  group_col = "sample_type",
  group_order = c("PNF", "ANNUBP", "MPNST"),
  reduced_dim = "TSNE",
  background = "solid"
)
p3
```

---

plotOverlayPerNestedGroup

*Plot overlays arranged by nested cell metadata groups*

---

## Description

plotOverlayPerNestedGroup creates one [plotOverlay](#) plot for each value of an inner grouping variable and arranges the plots according to a second, outer grouping variable. The usual use case is to create per-sample overlay plots and arrange them by sample type, condition or experimental group.

**Usage**

```

plotOverlayPerNestedGroup(
  sce,
  group_col,
  outer_col,
  foreground,
  background = "solid",
  reduced_dim = "TSNE",
  group_order = NULL,
  outer_order = NULL,
  fg_subset_cells = NULL,
  bg_type = c("auto", "continuous", "categorical"),
  bg_assay = "logcounts",
  fg_assay = "logcounts",
  fg_type = c("auto", "continuous", "categorical"),
  fg_order = c("input", "random", "ascending", "descending"),
  bg_point_size = 2,
  fg_point_size = 0.5,
  bg_palette = NULL,
  fg_palette = NULL,
  bg_raster = FALSE,
  fg_raster = FALSE,
  raster_dpi = 300,
  bg_dimming = 0.6,
  fg_limits = NULL,
  bg_limits = NULL,
  bg_legend = TRUE,
  fg_legend = TRUE,
  bg_legend_title = NULL,
  fg_legend_title = NULL,
  shared_legend = FALSE
)

```

**Arguments**

sce	A <a href="#">SingleCellExperiment</a> object containing the reduced dimensions and the data to plot. The object must contain the reduced dimension specified in <code>reduced_dim</code> , the grouping columns specified in <code>group_col</code> and <code>outer_col</code> , and the data required by foreground.
group_col	A single character value with the name of the <code>colData(sce)</code> column used to define the inner groups, usually the panels. This argument has no default and must be provided by the user.
outer_col	A single character value with the name of the <code>colData(sce)</code> column used to arrange the inner groups, usually as rows in the final plot. This argument has no default and must be provided by the user.
foreground	A single character value specifying the variable to draw in the foreground layer. It can be "solid" to draw selected foreground cells with a fixed colour, "none"

	to draw no visible foreground points, a feature in <code>rownames(sce)</code> , in which case values are taken from <code>fg_assay</code> , or a column in <code>colData(sce)</code> . This argument has no default and must be provided by the user.
<code>background</code>	A single character value specifying the background layer passed to <a href="#">plotOverlay</a> . It can be "solid" to draw all background cells with a fixed colour, "none" to draw no visible background points, a feature in <code>rownames(sce)</code> , or a column in <code>colData(sce)</code> . Defaults to "solid".
<code>reduced_dim</code>	A single character value with the name of the reduced dimension to plot. It must be one of <code>reducedDimNames(sce)</code> . Only the first two dimensions are used. Defaults to "TSNE".
<code>group_order</code>	Optional character vector specifying the order in which inner groups should be plotted. If NULL, factor levels are used for factor grouping variables; otherwise, the order of appearance in <code>group_col</code> is used. Defaults to NULL.
<code>outer_order</code>	Optional character vector specifying the order of the outer groups. If NULL, factor levels are used for factor grouping variables; otherwise, the order of appearance in <code>outer_col</code> is used. Defaults to NULL.
<code>fg_subset_cells</code>	An optional additional cell subset applied to the foreground layer before splitting by group. It can be NULL, a logical vector of length <code>ncol(sce)</code> , a character vector with cell names, a function receiving <code>sce</code> and returning a logical vector, or an expression evaluated in <code>colData(sce)</code> . NULL selects all cells. NA values are treated as FALSE. Defaults to NULL. Inner groups with no matching foreground cells still produce panels with empty foreground layers, preserving the nested layout.
<code>bg_type</code>	A character value indicating how the background variable should be represented when background is not "solid" or "none". Accepted values are "auto", "continuous" and "categorical". See <a href="#">plotOverlay</a> for details. Defaults to "auto".
<code>bg_assay</code>	A single character value with the name of the assay to use when background is a feature in <code>rownames(sce)</code> . It is ignored when <code>background = "solid"</code> or <code>background = "none"</code> . Defaults to "logcounts".
<code>fg_assay</code>	A single character value with the name of the assay to use when foreground is a feature in <code>rownames(sce)</code> . It is ignored when <code>foreground = "solid"</code> or <code>foreground = "none"</code> . Defaults to "logcounts".
<code>fg_type</code>	A character value indicating how the foreground variable should be represented when foreground is not "solid" or "none". Accepted values are "auto", "continuous" and "categorical". With "auto", feature values are treated as continuous and <code>colData</code> variables are treated as continuous only when they are numeric. Defaults to "auto".
<code>fg_order</code>	A character value controlling the drawing order of foreground cells within each panel. Accepted values are "input", "random", "ascending" and "descending". See <a href="#">plotOverlay</a> for details. Defaults to "input".
<code>bg_point_size</code>	Numeric value with the point size of the background layer in each panel. Defaults to 2.
<code>fg_point_size</code>	Numeric value with the point size of the foreground layer in each panel. Defaults to 0.5.

<code>bg_palette</code>	Palette specification for the background layer. It can be NULL, a palette name, a vector of colours or a palette function. When <code>background = "solid"</code> , NULL uses "gray70" and a single valid colour can be used to set the solid background colour. It is ignored when <code>background = "none"</code> . For variable backgrounds, the value is passed to <code>getPalette</code> . Defaults to NULL, which uses "gray_red" for continuous backgrounds and "scOverlay" for categorical backgrounds.
<code>fg_palette</code>	Palette specification for the foreground layer. When <code>foreground = "solid"</code> , NULL uses "red" and a single valid colour can be used to set the solid foreground colour. It is ignored when <code>foreground = "none"</code> . For variable foregrounds, it can be NULL, a palette name, a vector of colours or a palette function. The value is passed to <code>getPalette</code> . For continuous foregrounds, colour vectors are interpreted as gradient anchors and interpolated internally. Defaults to NULL, which uses "gray_red" for continuous foregrounds and "scOverlay" for categorical foregrounds.
<code>bg_raster</code>	Logical value. If TRUE, rasterize the background point layer in each panel using <code>ggrastr::geom_point_rast()</code> . Defaults to FALSE.
<code>fg_raster</code>	Logical value. If TRUE, rasterize the foreground point layer in each panel using <code>ggrastr::geom_point_rast()</code> . Defaults to FALSE.
<code>raster_dpi</code>	Positive numeric value with the resolution, in dots per inch, used for rasterized point layers. Defaults to 300.
<code>bg_dimming</code>	Numeric value between 0 and 1 controlling the opacity of the white dimming layer placed between background and foreground in each panel. A value of 0 does not dim the background and a value of 1 completely covers it with white. Defaults to 0.6.
<code>fg_limits</code>	Limits for the foreground colour scale when the foreground is continuous. It can be NULL, a strictly increasing numeric vector of length two, or "shared". With "shared", one range is computed after the global <code>fg_subset_cells</code> filter and reused in every panel. Defaults to NULL.
<code>bg_limits</code>	Limits for the background colour scale when the background is continuous. It can be NULL, a strictly increasing numeric vector of length two, or "shared". With "shared", one range is computed over all cells and reused in every panel. Defaults to NULL.
<code>bg_legend</code>	Logical value indicating whether to show the background legend in each panel. It is ignored when <code>background = "solid"</code> . Defaults to TRUE.
<code>fg_legend</code>	Logical value indicating whether to show the foreground legend in each panel. It is ignored when <code>foreground = "solid"</code> . Defaults to TRUE.
<code>bg_legend_title</code>	Character value with the title of the background legend. If NULL, background is used. Defaults to NULL.
<code>fg_legend_title</code>	Character value with the title of the foreground legend. If NULL, foreground is used. Defaults to NULL.
<code>shared_legend</code>	Logical value indicating whether compatible legends should be collected into a shared patchwork legend. Defaults to FALSE.

## Details

This function is useful when the cells belong to two related annotations, for example samples nested within sample types. Each panel keeps the full embedding as background, while the foreground layer is restricted to the cells of the corresponding inner group.

The inner grouping variable is specified with `group_col` and defines the individual panels. The outer grouping variable is specified with `outer_col` and is used to arrange these panels in rows. For example, if `group_col = "sample"` and `outer_col = "sample_type"`, each panel will show one sample and panels from the same sample type will be placed in the same row.

For each inner group, `plotOverlayPerNestedGroup` calls `plotOverlay` using the same background, foreground, palette and plotting options. The foreground subset for each panel is defined by the inner group membership and, optionally, by the additional filter supplied with `fg_subset_cells`.

As in `plotOverlay`, subsetting only affects the foreground layer. Unless `background = "none"`, the background layer still shows all cells in every panel, providing a common reference embedding across all groups. This makes it possible to compare where samples or other inner groups are located in the same reduced dimension space, while keeping the broader outer grouping visible in the layout.

With `fg_limits = "shared"`, one foreground range is computed after the global `fg_subset_cells` filter and before splitting into panels. With `bg_limits = "shared"`, one background range is computed across all cells. Categorical foreground and background palettes are based on global categorical values, so colours remain consistent across panels. Named categorical palettes are recommended when strict category-colour control is needed.

If `shared_legend = TRUE`, compatible legends are collected with `patchwork`. The legend can be repositioned afterwards with `patchwork` syntax, for example `p & ggplot2::theme(legend.position = "bottom")`.

The function assumes that the inner groups are meaningfully associated with the outer groups. In the typical case, each value of `group_col` belongs to a single value of `outer_col`.

If `group_order` or `outer_order` are provided, panels and rows are drawn following those orders. If they are `NULL`, factor levels are used when the corresponding grouping variable is a factor; otherwise, values are shown in their order of appearance in the object.

## Value

A `patchwork` object arranging one `plotOverlay` plot per inner group and outer group. If no valid plots can be generated, the function returns `NULL` with a warning.

## See Also

[plotOverlay](#), [plotOverlayPerGroup](#), [plotGeneOverlay](#), [getPalette](#) and [listPalettes](#).

## Examples

```
data("sce_overlay_example")

# Plot SOX10 expression by sample, arranging samples by sample type.
p1 <- plotOverlayPerNestedGroup(
  sce = sce_overlay_example,
  foreground = "SOX10",
```

```

    group_col = "sample",
    outer_col = "sample_type",
    reduced_dim = "TSNE",
    background = "solid",
    fg_order = "ascending"
  )
  p1

# Use the cluster annotation as background in every panel.
p2 <- plotOverlayPerNestedGroup(
  sce = sce_overlay_example,
  foreground = "S100B",
  group_col = "sample",
  outer_col = "sample_type",
  reduced_dim = "TSNE",
  background = "cluster",
  fg_order = "ascending"
)
p2

# Specify explicit orders for rows and panels.
p3 <- plotOverlayPerNestedGroup(
  sce = sce_overlay_example,
  foreground = "SOX10",
  group_col = "sample",
  outer_col = "sample_type",
  group_order = c("S1", "S2", "S3", "S4", "S5", "S6"),
  outer_order = c("PNF", "ANNUBP", "MPNST"),
  reduced_dim = "TSNE",
  background = "solid"
)
p3

```

---

 sce\_mpnst\_example

*Example MPNST SingleCellExperiment for the scOverlay vignette*


---

## Description

A small [SingleCellExperiment](#) object derived from the single-cell MPNST progression dataset deposited at the European Genome-phenome Archive under accession EGAS50000001747. The object has been reduced in size and is included only to demonstrate `scOverlay` functionality in the package vignette.

## Format

A [SingleCellExperiment](#) object with selected genes, cell metadata and reduced dimensions.

**Source**

European Genome-phenome Archive study EGAS50000001747, <https://ega-archive.org/studies/EGAS50000001747>.

**Examples**

```
data("sce_mpnst_example")
sce_mpnst_example
```

---

sce\_overlay\_example    *Small example SingleCellExperiment for scOverlay*

---

**Description**

A small [SingleCellExperiment](#) object used in scOverlay examples and tests. It contains a reduced sintetic set of cells, genes, cell annotations and reduced dimensions to demonstrate the basic plotting functions quickly. It includes logcounts, TSNE and UMAP reduced dimensions, and cell metadata such as sample, sample\_type and cluster.

**Format**

A [SingleCellExperiment](#) object with gene expression values, cell metadata and reduced dimensions.

**Examples**

```
data("sce_overlay_example")
sce_overlay_example
```

# Index

## \* datasets

sce\_mpnst\_example, [20](#)

sce\_overlay\_example, [21](#)

getPalette, [2](#), [5](#), [6](#), [8](#), [10](#), [13](#), [14](#), [18](#), [19](#)

listPalettes, [3](#), [4](#), [6](#), [10](#), [14](#), [19](#)

plotGeneOverlay, [5](#), [10](#), [14](#), [19](#)

plotOverlay, [3](#), [5](#), [6](#), [6](#), [11–15](#), [17](#), [19](#)

plotOverlayPerGroup, [10](#), [11](#), [19](#)

plotOverlayPerNestedGroup, [10](#), [14](#), [15](#)

sce\_mpnst\_example, [20](#)

sce\_overlay\_example, [21](#)

SingleCellExperiment, [5–7](#), [12](#), [16](#), [20](#), [21](#)