Package ‘ggsc’

March 20, 2024

Title  Visualizing Single Cell Data
Version  1.0.2
Description  Useful functions to visualize single cell and spatial data. It supports both 'SingleCellExperiment' and 'Seurat' objects. It also supports visualizing the data using grammar of graphics implemented in 'ggplot2'.
Imports  Rcpp, RcppParallel, cli, dplyr, ggplot2, grDevices, grid, methods, rlang, scattermore, stats, Seurat, SingleCellExperiment, SummarizedExperiment, tidydr, tidyr, tibble, utils, yulab.utils
Suggests  aplot, BiocParallel, forcats, ggforce, ggnewscale, igraph, knitr, ks, Matrix, prettydoc, rmarkdown, scran, scater, scuttle, shadowtext, sf, SeuratObject, SpatialExperiment, STexampleData, testthat (>= 3.0.0)
BugReports  https://github.com/YuLab-SMU/ggsc/issues
URL  https://github.com/YuLab-SMU/ggsc
biocViews  DimensionReduction, GeneExpression, SingleCell, Software, Spatial, Transcriptomics, Visualization
VignetteBuilder  knitr
ByteCompile  true
License  Artistic-2.0
Encoding  UTF-8
Roxygen  list(markdown = TRUE)
RoxygenNote  7.2.3
Config/testthat/edition  3
LinkingTo  Rcpp, RcppArmadillo, RcppParallel
git_url  https://git.bioconductor.org/packages/ggsc
git_branch  RELEASE_3_18
git_last_commit  67f8871
git_last_commit_date  2023-10-31
Description

Useful functions to visualize single cell and spatial data. It supports both 'SingleCellExperiment' and 'Seurat' objects. It also supports visualizing the data using grammar of graphics implemented in 'ggplot2'.

Author(s)

Maintainer: Guangchuang Yu <guangchuangyu@gmail.com> (ORCID) [copyright holder]
Authors:

- Shuangbin Xu <xshuangbin@163.com> (ORCID)

See Also

Useful links:

- https://github.com/YuLab-SMU/ggsc
- Report bugs at https://github.com/YuLab-SMU/ggsc/issues
Two-Dimensional Weighted Kernel Density Estimation And Mapping the Result To Original Dimension

Usage

CalWkdeCpp(x, w, l, h, adjust = 1, n = 400L)

Arguments

- **x**: The 2-D coordinate matrix
- **w**: The weighted sparse matrix, the number columns the same than the number rows than x.
- **l**: The limits of the rectangle covered by the grid as c(xl, xu, yl, yu)
- **h**: The vector of bandwidths for x and y directions, defaults to normal reference bandwidth (see bandwidth.nrd), A scalar value will be taken to apply to both directions (see ks::hpi).
- **adjust**: numeric value to adjust to bandwidth, default is 1.
- **n**: number of grid points in the two directions, default is 400.

Description

These objects are imported from other packages. Follow the links below to see their documentation.

- **ggplot2**
  - aes, theme

Value

Depending on the re-exported function
sc_dim

Description
sc_dim

Usage
sc_dim(
  object,
  dims = c(1, 2),
  reduction = NULL,
  cells = NULL,
  slot = "data",
  mapping = NULL,
  ...
)

## S4 method for signature 'Seurat'
sc_dim(
  object,
  dims = c(1, 2),
  reduction = NULL,
  cells = NULL,
  slot = "data",
  mapping = NULL,
  ...
)

## S4 method for signature 'SingleCellExperiment'
sc_dim(
  object,
  dims = c(1, 2),
  reduction = NULL,
  cells = NULL,
  slot = "data",
  mapping = NULL,
  ...
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>Seurat object</td>
</tr>
<tr>
<td>dims</td>
<td>selected dimensions (must be a two-length vector) that are used in visualization</td>
</tr>
<tr>
<td>reduction</td>
<td>reduction method, default is NULL and will use the default setting store in the object</td>
</tr>
</tbody>
</table>
sc_dim_count

cells selected cells to plot (default is all cells)
slot slot to pull expression data from (e.g., 'count' or 'data')
mapping aesthetic mapping
... additional parameters pass to 'scattermore::geom_scattermore()'

Value
dimension reduction plot

See Also
gemm_scattermore;

Examples
library(scuttle)
library(scater)
library(scran)
library(ggplot2)
sce <- mockSCE()
sce <- logNormCounts(sce)
clusters <- clusterCells(sce, assay.type = 'logcounts')
colLabels(sce) <- clusters
sce <- runUMAP(sce, assay.type = 'logcounts')
p1 <- sc_dim(sce, reduction = 'UMAP', mapping = aes(colour = Cell_Cycle))
p2 <- sc_dim(sce, reduction = 'UMAP')
f1 <- p1 + sc_dim_geom_label()
f2 <- p2 +
    sc_dim_geom_label(
        geom = shadowtext::geom_shadowtext,
        color='black',
        bg.color='white'
    )
Value

a bar plot to present the cell numbers of different clusters

See Also

sc_dim()

Examples

library(scuttle)
library(scater)
library(scran)
library(ggplot2)
sce <- mockSCE()
sce <- logNormCounts(sce)
clusters <- clusterCells(sce, assay.type = 'logcounts')
colLabels(sce) <- clusters
sce <- runUMAP(sce, assay.type = 'logcounts')
p <- sc_dim(sce, reduction = 'UMAP')
p1 <- sc_dim_count(p)

sc_dim_geom_ellipse

Description

sc_dim_geom_ellipse

Usage

sc_dim_geom_ellipse(mapping = NULL, level = 0.95, ...)

Arguments

mapping aesthetic mapping
level the level at which to draw an ellipse
... additional parameters pass to the stat_ellipse

Value

layer of ellipse

See Also

stat_ellipse;
Examples

library(scuttle)
library(scater)
library(scran)
library(ggplot2)
sce <- mockSCE()
sce <- logNormCounts(sce)
clusters <- clusterCells(sce, assay.type = 'logcounts')
colLabels(sce) <- clusters
sce <- runUMAP(sce, assay.type = 'logcounts')
p1 <- sc_dim(sce, reduction = 'UMAP', mapping = aes(colour = Cell_Cycle))
p2 <- sc_dim(sce, reduction = 'UMAP')
f1 <- p1 + sc_dim_geom_ellipse()

Description

sc_dim_geom_feature

Usage

sc_dim_geom_feature(
  object,
  features,
  dims = c(1, 2),
  ncol = 3,
  ..., .fun = function(.data) dplyr::filter(.data, .data$value > 0)
)

Arguments

  object  Seurat or SingleCellExperiment object
  features selected features (i.e., genes)
  dims    selected dimensions (must be a two-length vector) that are used in visualization
  ncol    number of facet columns if `length(features) > 1`
  ...     additional parameters pass to `scattermore::geom_scattermore()`
  .fun    user defined function that will be applied to selected features (default is to filter
           out genes with no expression values)

Value

layer of points for selected features
See Also

`sc_feature()`

Examples

```r
library(scuttle)
library(scater)
library(scran)
library(ggplot2)
sce <- mockSCE()
sce <- logNormCounts(sce)
clusters <- clusterCells(sce, assay.type = 'logcounts')
colLabels(sce) <- clusters
sce <- runUMAP(sce, assay.type = 'logcounts')
p1 <- sc_dim(sce, reduction = 'UMAP')
set.seed(123)
genes <- rownames(sce) |> sample(6)
f1 <- p1 +
  sc_dim_geom_feature(
    object = sce,
    features = genes
  )
```

---

desc

Description

`sc_dim_geom_label`

Usage

`sc_dim_geom_label(geom = ggplot2::geom_text, ...)`

Arguments

- `geom`: geometric layer (default: `geom_text`) to display the labels
- `...`: additional parameters pass to the `geom`

Value

layer of labels

See Also

`sc_dim_geom_label()`
Examples

library(scuttle)
library(scater)
library(scran)
library(ggplot2)
sce <- mockSCE()
sce <- logNormCounts(sce)
clusters <- clusterCells(sce, assay.type = 'logcounts')
collabels(sce) <- clusters
sce <- runUMAP(sce, assay.type = 'logcounts')
p1 <- sc_dim(sce, reduction = 'UMAP', mapping = aes(colour = Cell_Cycle))
p2 <- sc_dim(sce, reduction = 'UMAP')
f1 <- p1 + sc_dim_geom_label()

sc_dim_geom_subset

Description

sc_dim_geom_subset

Usage

sc_dim_geom_subset(mapping = NULL, subset, .column = "ident", ...)

Arguments

mapping       aesthetic mapping
subset        subset of clusters to be displayed
.column       which column represents cluster (e.g., 'ident')
...            additional parameters pass to sc_geom_point

Value

plot with a layer of specified clusters

See Also

sc_dim_geom_sub

Examples

library(scuttle)
library(scater)
library(scran)
library(ggplot2)
sce <- mockSCE()
sce <- logNormCounts(sce)
clusters <- clusterCells(sce, assay.type = 'logcounts')
collLabels(sce) <- clusters
sce <- runUMAP(sce, assay.type = 'logcounts')
p1 <- sc_dim(sce, reduction = 'UMAP')
f1 <- p1 + sc_dim_geom_sub(subset = c(1, 2), .column = 'label')

---

### Description

`sc_dim_sub`

### Usage

```r
sc_dim_sub(subset, .column = "ident")
```

### Arguments

- **subset**: subset of clusters to be displayed
- **.column**: which column represents cluster (e.g., 'ident')

### Value

update plot with only subset displayed

### See Also

`sc_dim`

### Examples

```r
library(scuttle)
library(scater)
library(scran)
library(ggplot2)
sce <- mockSCE()
sce <- logNormCounts(sce)
clusters <- clusterCells(sce, assay.type = 'logcounts')
collLabels(sce) <- clusters
sce <- runUMAP(sce, assay.type = 'logcounts')
p1 <- sc_dim(sce, reduction = 'UMAP')
f1 <- p1 + sc_dim_sub(subset = c(1, 2), .column = 'label')
```
Description

sc_feature

Usage

sc_feature(
  object,
  features,
  dims = c(1, 2),
  reduction = NULL,
  cells = NULL,
  slot = "data",
  mapping = NULL,
  ncol = 3,
  density = FALSE,
  grid.n = 100,
  joint = FALSE,
  joint.fun = prod,
  common.legend = TRUE,
  ...
)

## S4 method for signature 'Seurat'
sc_feature(
  object,
  features,
  dims = c(1, 2),
  reduction = NULL,
  cells = NULL,
  slot = "data",
  mapping = NULL,
  ncol = 3,
  density = FALSE,
  grid.n = 100,
  joint = FALSE,
  joint.fun = prod,
  common.legend = TRUE,
  ...
)

## S4 method for signature 'SingleCellExperiment'
sc_feature(
  object,
features,  
dims = c(1, 2),  
reduction = NULL,  
cells = NULL,  
slot = "data",  
mapping = NULL,  
ncol = 3,  
density = FALSE,  
grid.n = 100,  
joint = FALSE,  
joint.fun = prod,  
common.legend = TRUE,  
...  
)

Arguments

object Seurat object
features selected features (i.e., genes)
dims selected dimensions (must be a two-length vector) that are used in visualization
reduction reduction method, default is NULL and will use the default setting store in the object
cells selected cells to plot (default is all cells)
slot slot to pull expression data from (e.g., 'count' or 'data')
mapping aesthetic mapping
ncol number of facet columns if 'length(features) > 1'
density whether plot the 2D weighted kernel density, default is FALSE.
grid.n number of grid points in the two directions to estimate 2D weighted kernel density, default is 100.
joint whether joint the multiple features with joint.fun, default is FALSE.
joint.fun how to joint the multiple features if joint=TRUE, default is prod.
common.legend whether to use facet_wrap to display the multiple features, default is TRUE.
... additional parameters pass to 'scattermore::geom_scattermore()'

Value
dimension reduction plot colored by selected features

Examples

library(scuttle)
library(scater)
library(scran)
library(ggplot2)
sce <- mockSCE()
sce <- logNormCounts(sce)
clusters <- clusterCells(sce, assay.type = 'logcounts')
colLabels(sce) <- clusters
sce <- runTSNE(sce, assay.type = 'logcounts')
set.seed(123)
genes <- rownames(sce) |> sample(6)
p1 <- sc_feature(sce, genes[1], slot='logcounts', reduction = 'TSNE')
p2 <- sc_feature(sce, genes, slot='logcounts', reduction = 'TSNE')
f1 <- sc_dim(sce, slot='logcounts', reduction = 'TSNE') +
    sc_dim_geom_feature(sce, genes[1], color='black')
f2 <- sc_dim(sce, alpha=.3, slot='logcounts', reduction = 'TSNE') +
ggnewscale::new_scale_color() +
    sc_dim_geom_feature(sce, genes, mapping=aes(color=features)) +
    scale_color_viridis_d()
p1 + p2 + f1 + f2

sc_geom_point

Description

sc_geom_point

Usage

sc_geom_point(mapping = NULL, ...)

Arguments

mapping aesesthetic mapping

... additional parameters pass to 'scattermore::geom_scattermore()'

Value

layer of points

See Also

sc_dim() and sc_feature()

Examples

library(ggplot2)
ggplot(iris,
    aes(x= Sepal.Length, y = Petal.Width, color=Species)
  ) +
sc_geom_point()
Description

`sc_spatial`

Usage

```r
sc_spatial(
  object,
  features = NULL,
  sample.id = NULL,
  image.id = NULL,
  slot = "data",
  image.plot = TRUE,
  image.first.operation = "rotate",
  image.rotate.degree = NULL,
  image.mirror.axis = NULL,
  remove.point = FALSE,
  mapping = NULL,
  ncol = 6,
  density = FALSE,
  grid.n = 100,
  joint = FALSE,
  joint.fun = prod,
  common.legend = TRUE,
  point.size = 5,
  ...
)
```

```r
## S4 method for signature 'Seurat'
sc_spatial(
  object,
  features = NULL,
  sample.id = NULL,
  image.id = NULL,
  slot = "data",
  image.plot = TRUE,
  image.first.operation = "rotate",
  image.rotate.degree = NULL,
  image.mirror.axis = NULL,
  remove.point = FALSE,
  mapping = NULL,
  ncol = 6,
  density = FALSE,
  grid.n = 100,
)```

Arguments

object  Seurat object
features selected features to be visualized
sample.id the index name of sample id, which only work with SingleCellExperiment or SpatialExperiment.
image.id the index name of image id, which only work with SingleCellExperiment or SpatialExperiment.
slot if plotting a feature, which data will be used (e.g., 'data', 'counts'), the assay name if object is SingleCellExperiment or SpatialExperiment.
image.plot whether to display the issue image as background.
image.first.operation character which the first operation to image, 'rotate' or 'mirror', default is 'rotate'.
image.rotate.degree integer the degree to ratate image, default is NULL.
image.mirror.axis
character the direction to mirror the image, default is 'h'.

remove.point
whether to remove the spot points, it is nice if your just view the issue image, default is FALSE.

mapping
aesthetic mapping, default is NULL.

ncol
integer number of facet columns if 'length(features) > 1', default is 6.

density
whether plot the 2D weighted kernel density, default is FALSE.

grid.n
number of grid points in the two directions to estimate 2D weighted kernel density, default is 100.

joint
whether joint the multiple features with joint.fun, default is FALSE.

joint.fun
how to joint the multiple features if joint = TRUE, default is prod.

common.legend
whether to use facet_wrap to display the multiple features, default is TRUE.

point.size
the size of point, default is 5.

... additional parameters.

Value

ggplot object

Examples

```r
## Not run:
library(STexampleData)
# create ExperimentHub instance
eh <- ExperimentHub()
# query STexampleData datasets
myfiles <- query(eh, "STexampleData")
spe <- myfiles[["EH7538"]]
spe <- spe[, colData(spe)$in_tissue == 1]
set.seed(123)
gen <- rownames(spe) |> sample(6)
p <- sc.spatial(spe, features = genes, 
                image.rotate.degree = -90, 
                image.mirror.axis = NULL, 
                ncol = 3)

## End(Not run)
```

sc_violin

Description

sc_violin
Usage

```r
sc_violin(
  object,
  features,
  cells = NULL,
  slot = "data",
  .fun = NULL,
  mapping = NULL,
  ncol = 3,
  ...
)
```

```r
## S4 method for signature 'Seurat'
sc_violin(
  object,
  features,
  cells = NULL,
  slot = "data",
  .fun = NULL,
  mapping = NULL,
  ncol = 3,
  ...
)
```

```r
## S4 method for signature 'SingleCellExperiment'
sc_violin(
  object,
  features,
  cells = NULL,
  slot = "data",
  .fun = NULL,
  mapping = NULL,
  ncol = 3,
  ...
)
```

Arguments

- **object**: Seurat object
- **features**: selected features
- **cells**: selected cells to plot (default is all cells)
- **slot**: slot to pull expression data from (e.g., 'count' or 'data')
- **.fun**: user defined function that will be applied to selected features (default is NULL and there is no data operation)
- **mapping**: aesthetic mapping
- **ncol**: number of facet columns if `length(features) > 1`
- **...**: additional parameters pass to `ggplot2::geom_geom_violin()`
**Value**

violin plot to visualize feature expression distribution

**See Also**

`geom_violin`;

**Examples**

```r
library(scuttle)
library(scater)
library(scran)
library(ggplot2)
sce <- mockSCE()
sce <- logNormCounts(sce)
clusters <- clusterCells(sce, assay.type = 'logcounts')
colLabels(sce) <- clusters
sce <- runUMAP(sce, assay.type = 'logcounts')
set.seed(123)
genes <- rownames(sce) |> sample(6)
sc_violin(sce, genes[1], slot = 'logcounts')
sc_violin(sce, genes[1], slot = 'logcounts',
  .fun=function(d) dplyr::filter(d, value > 0)
) +
  ggforce::geom_sina(size=.1)
sc_violin(sce, genes, slot = 'logcounts') +
  theme(axis.text.x = element_text(angle=45, hjust=1))
```
Index

* internal
  ggsc-package, 2
  reexports, 3

aes, 3
aes(reexports), 3

CalWkdeCpp, 3

ggsc-package, 2

geom_scattermore, 5
geom_violin, 18
ggsc(ggsc-package), 2

reexports, 3

sc_dim, 4, 10
sc_dim(), 6, 13
sc_dim, Seurat(sc_dim), 4
sc_dim, Seurat-method(sc_dim), 4
sc_dim, SingleCellExperiment(sc_dim), 4
sc_dim, SingleCellExperiment-method(sc_dim), 4
sc_dim_count, 5
sc_dim_geom_ellipse, 6
sc_dim_geom_feature, 7
sc_dim_geom_label, 8
sc_dim_geom_label(), 8
sc_dim_geom_sub, 9, 9
sc_dim_sub, 10
sc_feature, 11
sc_feature(), 8, 13
sc_feature, Seurat(sc_feature), 11
sc_feature, Seurat-method(sc_feature), 11
sc_feature, SingleCellExperiment(sc_feature), 11
sc_feature, SingleCellExperiment-method(sc_feature), 11
sc_geom_point, 13
scSpatial, 14

scSpatial, Seurat(scSpatial), 14
scSpatial, Seurat-method(scSpatial), 14
scSpatial, SingleCellExperiment(scSpatial), 14
scSpatial, SingleCellExperiment-method(scSpatial), 14
sc_violin, 16
sc_violin, Seurat(sc_violin), 16
sc_violin, Seurat-method(sc_violin), 16
sc_violin, SingleCellExperiment(sc_violin), 16
sc_violin, SingleCellExperiment-method(sc_violin), 16

stat_ellipse, 6
theme, 3
theme(reexports), 3