Package 'MerfishData'

February 2, 2023

Title Collection of public MERFISH datasets

Version 1.0.0

Description MerfishData is an ExperimentHub package that serves publicly available datasets obtained with Multiplexed Error-Robust Fluorescence in situ Hybridization (MERFISH). MERFISH is a massively multiplexed single-molecule imaging technology capable of simultaneously measuring the copy number and spatial distribution of hundreds to tens of thousands of RNA species in individual cells. The scope of the package is to provide MERFISH data for benchmarking and analysis.

License Artistic-2.0

VignetteBuilder knitr

LazyData false

URL https://github.com/ccb-hms/MerfishData

BugReports https://github.com/ccb-hms/MerfishData/issues

Depends R (>= 4.2.0), EBImage, SpatialExperiment

Imports grDevices, AnnotationHub, BumpyMatrix, ExperimentHub, S4Vectors, SummarizedExperiment

Suggests grid, ggplot2, ggpubr, knitr, rmarkdown, testthat, BiocStyle, ExperimentHubData

biocViews ExperimentHub, ExpressionData, HighThroughputImagingData, Mus_musculus_Data, SingleCellData, SpatialData

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.0

git_url https://git.bioconductor.org/packages/MerfishData

git_branch RELEASE_3_16

git_last_commit 663ad93

git_last_commit_date 2022-11-01

Date/Publication 2023-02-02

Author Ludwig Geistlinger [aut, cre] (<https://orcid.org/0000-0002-2495-5464>),
Tyrone Lee [ctb],
Helena Crowell [ctb] (<https://orcid.org/0000-0002-4801-1767>),
Jeffrey Mofitt [aut],
Robert Gentleman [aut]

Maintainer Ludwig Geistlinger <ludwig_geistlinger@hms.harvard.edu>

R topics documented:

`MerfishData-package` *Collection of public MERFISH datasets*

Description

MerfishData is an ExperimentHub package that serves publicly available datasets obtained with Multiplexed Error-Robust Fluorescence in situ Hybridization (MERFISH). MERFISH is a massively multiplexed single-molecule imaging technology capable of simultaneously measuring the copy number and spatial distribution of hundreds to tens of thousands of RNA species in individual cells. The scope of the package is to provide MERFISH data for benchmarking and analysis.

Author(s)

Ludwig Geistlinger ludwig_geistlinger@hms.harvard.edu

`addHolesToPolygons` *Add holes to polygons*

Description

Add holes to a `data.frame` of cell polygon coordinates

Usage

```
addHolesToPolygons(poly)
```

Arguments

poly	A data.frame storing cell polygon coordinates. Expected columns include • "cell" storing the cell ID, • "x" storing x-coordinates of the polygon corners, • "y" storing the y-coordinates of the polygon corners.

Value

A data.frame

Examples

```
x <- c(2053, 2053, 2053, 2056, 2059, 2059)
y <- c(51, 54, 57, 57, 57, 54)
poly <- data.frame(cell = 1, x = x, y = y)
poly <- addHolesToPolygons(poly)
```

MouseHypothalamusMoffitt2018

MERFISH mouse hypothalamus dataset from Moffitt et al., 2018

Description

Obtain the MERFISH mouse hypothalamic preoptic region dataset from Moffitt et al., 2018

Usage

```
MouseHypothalamusMoffitt2018(center.coords = TRUE)
```

Arguments

center.coords	logical. Should spatial x- and y-coordinates be centered for each z-layer (bregma slice)? This is useful for making coordinates comparable between bregma slices for visualization and analysis. Defaults to TRUE. Use FALSE to obtain the coordinates as provided in the data release.

Details

The hypothalamus controls essential social behaviors and homeostatic functions. However, the cellular architecture of hypothalamic nuclei, including the molecular identity, spatial organization, and function of distinct cell types, is not well understood.

Moffitt et al., 2018, developed an imaging-based cell type identification and mapping method and combined it with single-cell RNA-sequencing to create a molecularly annotated and spatially resolved cell atlas of the mouse hypothalamic preoptic region.

Value

An object of class SpatialExperiment.

Source

https://doi.org/10.5061/dryad.8t8s248

References

Moffitt et al. (2018) Molecular, spatial, and functional single-cell profiling of the hypothalamic preoptic region. Science, 362(6416), eaau5324.

Examples

```
spe <- MouseHypothalamusMoffitt2018()
```

MouseIleumPetukhov2021

MERFISH mouse ileum dataset from Petukhov et al., 2021

Description

Obtain the MERFISH mouse ileum dataset from Petukhov et al., 2021

Usage

```
MouseIleumPetukhov2021(
  segmentation = c("baysor", "cellpose"),
  use.images = TRUE,
  use.polygons = TRUE
)
```

Arguments

segmentation	character. Should be either "baysor" or "cellpose". Defaults to "baysor". See details.
use.images	logical. Should DAPI and Membrane Na+/K+ - ATPase images be loaded into memory and annotated to the imgData slot of the returned SpatialExperiment? Defaults to TRUE. See details.
use.polygons	logical. Should polygon cell boundaries be annotated to the metadata of the returned SpatialExperiment? Defaults to TRUE. Only available for Baysor segmentation.

Details

Spatial transcriptomics protocols based on in situ sequencing or multiplexed RNA fluorescent hybridization can reveal detailed tissue organization. Distinguishing the boundaries of individual cells in such data is challenging. Current segmentation methods typically approximate cells positions using nuclei stains.

Petukhov et al., 2021, describe Baysor, a segmentation method, which optimizes 2D or 3D cell boundaries considering joint likelihood of transcriptional composition and cell morphology. Baysor can also perform segmentation based on the detected transcripts alone.

Petukhov et al., 2021, compare the results of Baysor segmentation (mRNA-only) to the results of a deep learning-based segmentation method called Cellpose from Stringer et al., 2021. Cellpose applies a machine learning framework for the segmentation of cell bodies, membranes and nuclei from microscopy images.

The function allows to obtain segmented MERFISH mouse ileum data for both segmentation methods.

A note on storing images within a `SpatialExperiment`: The default `use.images = TRUE` reduces the 9-frame z-stack images for DAPI stain and Membrane Na+/K+ - ATPase fluorecense to single-frame images (taking the first frame). For working with the 9-frame z-stack images it is recommended to load the images individually from ExperimentHub.

Value

An object of class `SpatialExperiment`.

Source

https://doi.org/10.5061/dryad.jm63xsjb2

References

Petukhov et al. (2021) Cell segmentation in imaging-based spatial transcriptomics. Nat Biotechnol, 40(3), 345-54.

Stringer et al. (2021) Cellpose: a generalist algorithm for cellular segmentation. Nat Methods, 18(1), 100-6.

Examples

```
spe <- MouseIleumPetukhov2021()
```

---

plotRasterImage *Plot raster image*

Description

Small helper function to plot a raster image.

Usage

```
plotRasterImage(img)
```

Arguments

img	a raster object representing a bitmap image.

Value

A ggplot object.

Examples

```
hgrid <- hcl(0, 80, seq(50, 80, 10))
img <- as.raster(matrix(hgrid, nrow = 4, ncol = 5))
plotRasterImage(img)
```

---

plotTabset *Plot a tabset*

Description

Plot a tabset of colData annotations of one or more SpatialExperiment objects over an image.

Usage

```
plotTabset(spe.list, img)
```

Arguments

spe.list	A named list of SpatialExperiment objects.
img	a raster object representing a bitmap image.

Value

None. Produces a tabset for rendering with rmarkdown.

Examples

```
library(SpatialExperiment)
# create simulated data as described in the SpatialExperiment man page
n <- 1000; ng <- 50; nc <- 20
# sample xy-coordinates in [0,1]
x <- runif(n)
y <- runif(n)
# assign each molecule to some gene-cell pair
gs <- paste0("gene", seq(ng))
cs <- paste0("cell", seq(nc))
```

```
gene <- sample(gs, n, TRUE)
cell <- sample(cs, n, TRUE)
# construct data.frame of molecule coordinates
df <- data.frame(gene, cell, x, y)

# (assure gene & cell are factor so that
# missing observations aren't dropped)
df$gene <- factor(df$gene, gs)
df$cell <- factor(df$cell, cs)

# construct BumpyMatrix
mol <- BumpyMatrix::splitAsBumpyMatrix(
    df[, c("x", "y")],
    row = df$gene, column = df$cell)

# get count matrix
y <- with(df, table(gene, cell))
y <- as.matrix(unclass(y))

# construct SpatialExperiment
spe <- SpatialExperiment(assays = list(counts = y, molecules = mol))

# add simulated cell centroids
s <- cbind(x = runif(20), y = runif(20))
spatialCoords(spe) <- s

# add simulated cell type and cell cycle annotation
ct <- c("ct1", "ct2", "ct3")
cc <- c("G1", "G2", "S", "M")
spe$type <- sample(ct, ncol(spe), replace = TRUE)
spe$cycle <- sample(cc, ncol(spe), replace = TRUE)

# create an example image
hgrid <- hcl(0, 80, seq(50, 80, 10))
img <- as.raster(matrix(hgrid, nrow = 4, ncol = 5))

# plotTabset
spe.list <- list(myseg = spe)
plotTabset(spe.list, img)
```

`plotXY` *Plot spatial image with data overlay*

Description

A helper function to overlay data onto a spatial image.

Usage

```
plotXY(df, col, img = NULL)
```

Arguments

df	A data.frame storing the data to plot.
col	character. A column of df to use for overlay onto the image.
img	a raster object representing a bitmap image.

Value

A ggplot.

Examples

```
gene <- rep(c("Cd44", "Cd8b1", "Cd79b"), each = 2)
x <- c(1693, 1701, 1820, 3188, 1631, 1881)
y <- c(1831, 1666, 1855, 6612, 1533, 942)
df <- data.frame(gene = gene, x = x, y = y)
plotXY(df, "gene")
```

Index