Package 'SpatialExperimentIO'

November 12, 2026

Title Read in Xenium, CosMx, MERSCOPE or STARmapPLUS data as SpatialExperiment object

Version 1.4.0

URL https://github.com/estellad/SpatialExperimentIO

BugReports https://github.com/estellad/SpatialExperimentIO/issues

Description Read in imaging-based spatial transcriptomics technology data. Current available modules are for Xenium by 10X Genomics, CosMx by Nanostring, MERSCOPE by Vizgen, or STARmapPLUS from Broad Institute. You can choose to read the data in as a SpatialExperiment or a SingleCellExperiment object.

License Artistic-2.0

Encoding UTF-8

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

Imports DropletUtils, SpatialExperiment, SingleCellExperiment, methods, data.table, arrow, purrr, S4Vectors

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

biocViews DataRepresentation, DataImport, Infrastructure, Transcriptomics, SingleCell, Spatial, GeneExpression

RoxygenNote 7.3.2

Config/testthat/edition 3

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

git_branch RELEASE_3_23

git_last_commit 4cd6a8b

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Yixing E. Dong [aut, cre] (ORCID: <https://orcid.org/0009-0003-5115-5686>)

Maintainer Yixing E. Dong <estelladong729@gmail.com>

Contents

.sanityCheck . . . 2
addParquetPathsCosmx . . . 3
addParquetPathsXenium . . . 4
addParquetPathToMeta . . . 6
csvToParquetPaths . . . 7
readCosmxSXE . . . 7
readMerscopeSXE . . . 9
readSeqfishSXE . . . 11
readStarmapplusSXE . . . 12
readXeniumSXE . . . 14

Index **17**

`.sanityCheck` *Sanity check if one and only file with the specified name pattern exists in the data download directory, and return the file path to .csv Used for count matrix and metadata only, as they require unique.*

Description

Sanity check if one and only file with the specified name pattern exists in the data download directory, and return the file path to .csv Used for count matrix and metadata only, as they require unique.

Usage

```
.sanityCheck(tech, filetype, expectfilename, dirName, filepatternvar)
```

Arguments

`tech`	Name of technology. Defined at the beginning of the function. e.g. "CosMx"
`filetype`	File type to do sanity check. e.g. "metadata"
`expectfilename`	Expected file pattern name for this file type. e.g. "metadata_file.csv"
`dirName`	Directory to the data download.
`filepatternvar`	The file pattern variable. e.g. "metaDataPattern"

Value

a path to a unique file of count matrix or colData.

Author(s)

Yixing Estella Dong

Examples

```
## Not run:
dir <- system.file(file.path("extdata", "CosMx_small"),
                   package = "SpatialExperimentIO")
countmat_file <- SpatialExperimentIO:::.sanityCheck(tech = "CosMx",
                              filetype = "count matrix",
                              expectfilename = "`exprMat_file.csv`",
                              dirName = dir,
                              filepatternvar = "exprMat_file.csv")

## End(Not run)
```

`addParquetPathsCosmx` *Add CosMx-related parquet paths to metadata for transcripts, polygon, or cell/nucleus boundaries.*

Description

Add CosMx-related parquet paths to metadata for transcripts, polygon, or cell/nucleus boundaries.

Usage

```
addParquetPathsCosmx(
  sxe,
  dirName,
  addTx = TRUE,
  txMetaNames = "transcripts",
  txPattern = "tx_file.csv",
  addPolygon = TRUE,
  polygonMetaNames = "polygons",
  polygonPattern = "polygons.csv"
)
```

Arguments

`sxe`	a SPE or SCE Xenium object to add parquet to `metadata(sxe)`.
`dirName`	the directory that stores the transcripts/polygon .csv or .parquet files.
`addTx`	to add path to transcripts parquet to `metadata(sxe)`or not. Default is TRUE.
`txMetaNames`	names to add to slots in `metadata(sxe)[["name"]]`. The number of `txMetaNames` should equal to number of file detected in `dirName` with `txPattern`. Can have multiple, such as `c("transcripts", "transcripts1")`. Default is `"transcripts"`.
`txPattern`	.csv or .parquet (if you have previous converted) pattern of transcript file in `dirName`. Can have multiple, such as `c("tx_file.csv", "tx_file1.csv")`. Default value is `"tx_file.csv"`.
`addPolygon`	to add path to polygons parquet to `metadata(sxe)` or not. Default is TRUE.

polygonMetaNames
: names to add to slots in `metadata(sxe)$`. The number of `polygonMetaNames` should equal to number of file detected in `dirName` with `polygonPattern`. Can have multiple. Can have multiple, such as `c("polygons", "polygons1")`. Default is `"transcripts"`.

polygonPattern
: .csv or .parquet (if you have previous converted) pattern of polygons file in the `dirName`. Can have multiple, such as `c("polygons.csv", "polygons1.csv")`. Default value is `"polygons.csv"`.

Value

a SPE or SCE Xenium object with parquet paths added to metadata

Author(s)

Yixing Estella Dong

Examples

```
cospath <- system.file(file.path("extdata", "CosMx_small"),
                       package = "SpatialExperimentIO")

sxe <- readCosmxSXE(dirName = cospath, addParquetPaths = FALSE)
sxe <- addParquetPathsCosmx(sxe, dirName = cospath, addPolygon = FALSE)
```

---

addParquetPathsXenium *Add Xenium-related parquet paths to metadata for transcripts or cell/nucleus boundaries.*

Description

Add Xenium-related parquet paths to metadata for transcripts or cell/nucleus boundaries.

Usage

```
addParquetPathsXenium(
  sxe,
  dirName,
  addTx = TRUE,
  txMetaNames = "transcripts",
  txPattern = "transcripts.parquet",
  addCellBound = TRUE,
  cellBoundMetaNames = "cell_boundaries",
  cellBoundPattern = "cell_boundaries.parquet",
  addNucBound = TRUE,
  NucBoundMetaNames = "nucleus_boundaries",
  NucBoundPattern = "nucleus_boundaries.parquet"
)
```

Arguments

sxe	a SPE or SCE Xenium object to add parquet to metadata(sxe).
dirName	the directory that stores the transcripts/cell_boundaries/nucleus_boundaries .parquet files.
addTx	to add path to transcripts parquet to metadata(sxe) or not. Default is FALSE.
txMetaNames	names to add to slots in metadata(sxe)[["name"]]. The number of txMetaNames should equal to number of file detected in dirName with txPattern. Can have multiple, such as c("transcripts", "transcripts1"). Default is "transcripts".
txPattern	.parquet pattern of transcript file in dirName. Can have multiple, such as c("transcripts.parquet", "transcripts1.parquet"). Default value is "transcripts.parquet".
addCellBound	to add path to cell boundaries parquet to metadata(sxe) or not. Default is FALSE.
cellBoundMetaNames	names to add to slots in metadata(sxe)[["name"]]. The number of cellBoundMetaNames should equal to number of file detected in dirName with cellBoundPattern. Can have multiple, such as c("cell_boundaries", "cell_boundaries1"). Default is "cell_boundaries".
cellBoundPattern	.parquet pattern of cell boundaries file in dirName. Can have multiple, such as c("cell_boundaries.parquet", "cell_boundaries1.parquet"). Default value is "cell_boundaries.parquet".
addNucBound	to add path to nucleus boundaries parquet to metadata(sxe) or not. Default is FALSE.
NucBoundMetaNames	names to add to slots in metadata(sxe)[["name"]]. The number of NucBoundMetaNames should equal to number of file detected in dirName with NucBoundPattern. Can have multiple, such as c("nucleus_boundaries", "nucleus_boundaries1"). Default is "nucleus_boundaries".
NucBoundPattern	.parquet pattern of nucleus boundaries file in dirName. Can have multiple, such as c("nucleus_boundaries.parquet", "nucleus_boundaries1.parquet"). Default value is "nucleus_boundaries.parquet".

Value

a SPE or SCE Xenium object with parquet paths added to metadata

Author(s)

Yixing Estella Dong

Examples

```
xepath <- system.file(file.path("extdata", "Xenium_small"),
                      package = "SpatialExperimentIO")

sxe <- readXeniumSXE(dirName = xepath, addParquetPaths = FALSE)
```

```
sxe <- addParquetPathsXenium(sxe, dirName = xepath)
```

`addParquetPathToMeta` *Add parquet paths to metadata for transcripts, polygon, or cell/nucleus boundaries.*

Description

Add parquet paths to metadata for transcripts, polygon, or cell/nucleus boundaries.

Usage

```
addParquetPathToMeta(
  sxe,
  dirName = dirName,
  metaNames = "transcripts",
  filePattern = "tx_file.csv"
)
```

Arguments

`sxe`	a SPE or SCE object to add parquet to `metadata(sxe)`.
`dirName`	the directory that stores the transcripts/polygon/cell_boundaries .csv or .parquet files.
`metaNames`	a vector of names to `metadata(sxe)[[]]`. The length must match number of files detected with filePattern provided. `e.g. c("transcripts", "transcripts1.csv")`.
`filePattern`	a vector of file patterns to search in the current directory. e.g. `c("tx_file.csv", "tx_file1.csv")`.

Value

a SPE or SCE object with parquet paths added to metadata

Examples

```
dir <- system.file(file.path("extdata", "CosMx_small"),
                   package = "SpatialExperimentIO")
sxe <- readCosmxSXE(dir, addParquetPaths = FALSE)
sxe <- addParquetPathToMeta(sxe,
                            dirName = dir,
                            metaNames = "transcripts",
                            filePattern = "tx_file.parquet")
```

`csvToParquetPaths` *If transcripts or polygon is expected to be loaded, write a parquet file to the current data download (if not already), and return the file path to .parquet*

Description

If transcripts or polygon is expected to be loaded, write a parquet file to the current data download (if not already), and return the file path to .parquet

Usage

```
csvToParquetPaths(dirName, filepath = "tx_csv_path")
```

Arguments

`dirName`	current directory of data download
`filepath`	path to transcripts or polygons csv

Value

a path to .parquet

Author(s)

Yixing Estella Dong

Examples

```
dir <- system.file(file.path("extdata", "CosMx_small"),
                   package = "SpatialExperimentIO")
tx_csv_path <- file.path(dir, "lung_p9s1_tx_file.csv")
tx_parquet_path <- csvToParquetPaths(dirName, filepath = tx_csv_path)
```

`readCosmxSXE` *Load data from a Nanostring CosMx experiment*

Description

Creates a `SpatialExperiment` from the downloaded unzipped CosMx directory for Nanostring CosMx spatial gene expression data.

Usage

```
readCosmxSXE(
  dirName = dirName,
  returnType = "SPE",
  countMatPattern = "exprMat_file.csv",
  metaDataPattern = "metadata_file.csv",
  coordNames = c("CenterX_global_px", "CenterY_global_px"),
  addFovPos = TRUE,
  fovPosPattern = "fov_positions_file.csv",
  altExps = c("NegPrb", "Negative", "SystemControl", "FalseCode"),
  addParquetPaths = TRUE,
  ...
)
```

Arguments

`dirName`	a directory path to CosMx download that contains files of interest.
`returnType`	option of `"SPE"` or `"SCE"`, stands for `SpatialExperiment` or `SingleCellExperiment` object. Default value `"SPE"`
`countMatPattern`	a filename pattern for the count matrix. Default value is `"exprMat_file.csv"`, and there is no need to change.
`metaDataPattern`	a filename pattern for the metadata .csv file that contains spatial coords. Default value is `"metadata_file.csv"`, and there is no need to change.
`coordNames`	a vector of two strings specify the spatial coord names. Default value is `c("CenterX_global_px", "CenterY_global_px")`, and there is no need to change.
`addFovPos`	to read in fov_position_list.csv and add the data frame to `metadata(sxe)$fov_positions` or not. Default is TRUE.
`fovPosPattern`	.csv pattern of fov_position_list.csv files in the raw download. Default value is `"fov_positions_file.csv"`.
`altExps`	gene names contains these strings will be moved to `altExps(sxe)` as separate sxe-s. Default is `c("NegPrb", "Negative", "SystemControl", "FalseCode")`.
`addParquetPaths`	to add parquet paths to `metadata(sxe)` or not. If TRUE, transcripts and polygon .csv files will be converted to .parquet, and the paths will be added. If, for instance, no polygon file is available, and only transcript file is available, please set this argument to TRUE and adjust `addPolygon = FALSE` in the `...` argument. Default is TRUE.
`...`	extra parameters to pass to `addParquetPathsCosmx()`, including `addTx`, `txMetaNames`, `txPattern`, `addPolygon`, `polygonMetaNames`, `polygonPattern`.

Details

The constructor assumes the downloaded unzipped CosMx folder has the following structure, with two mandatory files: CosMx_unzipped/optional_default_folder/

· | — *_exprMat_file.csv
· | — *_metadata_file.csv

Optional files to add to the metadata() as a list of paths (will be converted to parquet): · | — *_fov_positions_file.csv
· | — *_tx_file.csv
· | — *_polygons.csv
If no optional files, need to set `addFovPos = FALSE` and `addParquetPaths = FALSE`. If only one of `*_tx_file.csv` or `*_polygons.csv` exists, set `addParquetPaths = TRUE` but set the not available `addTx` or `addPolygon` to `FALSE`. See `addParquetPathsCosmx()`

Value

a `SpatialExperiment` or a `SingleCellExperiment` object

Author(s)

Yixing Estella Dong

Examples

```
# A relatively small data download can be from:
# https://nanostring.com/resources/smi-ffpe-dataset-lung9-rep1-data/


# A mock counts and mock metadata with spatial location generated for a 8 genes by
# 9 cells object is in /extdata:

cospath <- system.file(
  file.path("extdata", "CosMx_small"),
  package = "SpatialExperimentIO")

list.files(cospath)

# One of the following depending on your output (`SPE` or `SCE`) requirement.
cos_spe <- readCosmxSXE(dirName = cospath, addPolygon = FALSE)
cos_sce <- readCosmxSXE(dirName = cospath, returnType = "SCE", addPolygon = FALSE)
cos_spe <- readCosmxSXE(dirName = cospath, addParquetPaths = FALSE)
```

---

readMerscopeSXE *Load data from a Vizgen MERSCOPE experiment*

Description

Creates a `SpatialExperiment` from the downloaded MERSCOPE directory for Vizgen MERSCOPE spatial gene expression data.

Usage

```
readMerscopeSXE(
  dirName = dirName,
  returnType = "SPE",
  countMatPattern = "cell_by_gene.csv",
  metaDataPattern = "cell_metadata.csv",
  coordNames = c("center_x", "center_y")
)
```

Arguments

`dirName`	a directory path to MERSCOPE download that contains files of interest.
`returnType`	option of `"SPE"` or `"SCE"`, stands for `SpatialExperiment` or `SingleCellExperiment` object. Default value `"SPE"`
`countMatPattern`	a filename pattern for the count matrix. Default value is `"cell_by_gene.csv"`, and there is no need to change.
`metaDataPattern`	a filename pattern for the metadata .csv file that contains spatial coords. Default value is `"metadata_file.csv"`, and there is no need to change.
`coordNames`	a vector of two strings specify the spatial coord names. Default value is `c("center_x", "center_y")`, and there is no need to change.

Details

The constructor assumes the downloaded MERSCOPE count matrix and metadata in the same folder with the following structure: MERSCOPE_folder/
· | — cell_by_gene.csv
· | — cell_metadata.csv

Value

a `SpatialExperiment` or a `SingleCellExperiment` object

Author(s)

Yixing Estella Dong

Examples

```
# A relatively small data download can be from:
# https://console.cloud.google.com/storage/browser/vz-ffpe-showcase/
# HumanOvarianCancerPatient2Slice2?pageState=(%22StorageObjectListTable%22:
# (%22f%22:%22%255B%255D%22))&prefix=&forceOnObjectsSortingFiltering=false


# A mock counts and mock metadata with spatial location generated for a 9 genes by
# 8 cells object is in /extdata:
```

```
merpath <- system.file(
  file.path("extdata", "MERSCOPE_small"),
  package = "SpatialExperimentIO")

list.files(merpath)

# One of the following depending on your output (`SPE` or `SCE`) requirement.
mer_spe <- readMerscopeSXE(dirName = merpath)
mer_sce <- readMerscopeSXE(dirName = merpath, returnType = "SCE")
```

`readSeqfishSXE` *Load data from a Spatial Genomics seqFISH experiment*

Description

Creates a `SpatialExperiment` from the downloaded seqFISH directory for Spatial Genomics seqFISH spatial gene expression data.

Usage

```
readSeqfishSXE(
  dirName = dirName,
  returnType = "SPE",
  countMatPattern = "CellxGene.csv",
  metaDataPattern = "CellCoordinates.csv",
  coordNames = c("center_x", "center_y")
)
```

Arguments

`dirName`	a directory path to seqFISH download that contains files of interest.
`returnType`	option of `"SPE"` or `"SCE"`, stands for `SpatialExperiment` or `SingleCellExperiment` object. Default value `"SPE"`
`countMatPattern`	a filename pattern for the count matrix. Default value is `"CellxGene.csv"`, and there is no need to change.
`metaDataPattern`	a filename pattern for the metadata .csv file that contains spatial coords. Default value is `"CellCoordinates.csv"`, and there is no need to change.
`coordNames`	a vector of two strings specify the spatial coord names. Default value is `c("center_x", "center_y")`, and there is no need to change.

Details

The constructor assumes the downloaded seqFISH count matrix and metadata in the same folder with the following structure: seqFISH_folder/
· | — *_CellxGene.csv
· | — *_CellCoordinates.csv

Value

a SpatialExperiment or a SingleCellExperiment object

Author(s)

Yixing Estella Dong

Examples

```
# A relatively small data download can be from:
# https://spatialgenomics.com/data/#kidney-data


# A mock counts and mock metadata with spatial location generated for a 9 genes by
# 13 cells object is in /extdata:

seqfpath <- system.file(
  file.path("extdata", "seqFISH_small"),
  package = "SpatialExperimentIO")

list.files(seqfpath)

# One of the following depending on your output (`SPE` or `SCE`) requirement.
seqf_spe <- readSeqfishSXE(dirName = seqfpath)
seqf_sce <- readSeqfishSXE(dirName = seqfpath, returnType = "SCE")
```

---

readStarmapplusSXE *Load data from a STARmap PLUS experiment*

Description

Creates a SpatialExperiment from the downloaded STARmap PLUS count matrix.csv and metadata.csv

Usage

```
readStarmapplusSXE(
  dirName = dirName,
  returnType = "SPE",
  countMatPattern = "raw_expression_pd.csv",
  metaDataPattern = "spatial.csv",
  coordNames = c("X", "Y", "Z")
)
```

Arguments

dirName	a directory path to STARmap PLUS download that contains files of interest.
returnType	option of "SPE" or "SCE", stands for SpatialExperiment or SingleCellExperiment object. Default value "SPE"
countMatPattern	a filename pattern for the count matrix. Default value is "raw_expression_pd.csv", and there is no need to change.
metaDataPattern	a filename pattern for the metadata .csv file that contains spatial coords. Default value is "spatial.csv", and there is no need to change.
coordNames	a vector of three strings specify the spatial coord names. Default value is c("X", "Y", "Z"), and there is no need to change.

Details

The constructor assumes the downloaded unzipped STARmap PLUS folder has the following structure, with two mandatory files: STARmap_PLUS_download/
· | — *raw_expression_pd.csv
· | — *spatial.csv

Value

a SpatialExperiment or a SingleCellExperiment object

Author(s)

Yixing Estella Dong

Examples

```
# A relatively small data download can be from:
# https://zenodo.org/records/8327576


# A mock counts and mock metadata with spatial location generated for a 8 genes by
# 9 cells object is in /extdata:

starpath <- system.file(
```

```
  file.path("extdata", "STARmapPLUS_small"),
  package = "SpatialExperimentIO")

list.files(starpath)

# One of the following depending on your output (`SPE` or `SCE`) requirement.
star_spe <- readStarmapplusSXE(dirName = starpath)
star_sce <- readStarmapplusSXE(dirName = starpath, returnType = "SCE")
```

`readXeniumSXE` *Load data from a 10x Geonomics Xenium experiment*

Description

Creates a `SpatialExperiment` from the downloaded unzipped Xenium Output Bundle directory for 10x Genomics Xenium spatial gene expression data.

Usage

```
readXeniumSXE(
  dirName,
  returnType = "SPE",
  countMatPattern = "cell_feature_matrix.h5",
  metaDataPattern = "cells.parquet",
  coordNames = c("x_centroid", "y_centroid"),
  addExperimentXenium = TRUE,
 altExps = c("NegControlProbe", "UnassignedCodeword", "NegControlCodeword", "antisense",
    "BLANK"),
  addParquetPaths = TRUE,
  ...
)
```

Arguments

`dirName`	a directory path to Xenium Output Bundle download that contains files of interest.
`returnType`	option of `"SPE"` or `"SCE"`, stands for `SpatialExperiment` or `SingleCellExperiment` object. Default value `"SPE"`
`countMatPattern`	a folder directory or the h5 file pattern for the count matrix. Default value is `"cell_feature_matrix.h5"`, alternative value is `"cell_feature_matrix"` that takes a bit longer. The count matrix is read in and stored in a `SingleCellExperiment` object, using `DropletUtils::read10xCounts()`
`metaDataPattern`	a filename pattern of the zipped .csv file that contains spatial coords. Default value is `"cells.csv.gz"`, and there is no need to change.

`coordNames` a vector of two strings specify the spatial coord names. Default value is `c("x_centroid", "y_centroid")`, and there is no need to change.

`addExperimentXenium` to add experiment.xenium parameters to `metadata(sxe)` or not. Default value is TRUE.

`altExps` gene names contains these strings will be moved to `altExps(sxe)` as separate sxe-s. Default is `c("NegControlProbe", "UnassignedCodeword", "NegControlCodeword", "antisense", "BLANK")`.

`addParquetPaths` to add parquet paths to `metadata(sxe)` or not. If TRUE, transcripts, cell_boundaries, and nucleus_boundaries .parquet paths will be added to `metadata()`. If, for instance, no cell_boundaries file is available, and transcript and nucleus_boundaries files are available, please set this argument to TRUE and adjust `addCellBound = FALSE` in the `...` argument. Default is TRUE.

`...` extra parameters to pass to `addParquetPathsXenium()`, including `addTx`, `txMetaNames`, `txPattern`, `addCellBound`, `cellBoundMetaNames`, `cellBoundPattern`, `addNucBound`, `NucBoundMetaNames`, `NucBoundPattern`.

Details

The constructor assumes the downloaded unzipped Xenium Output Bundle has the following structure, with mandatory file of cells.csv.gz and either folder /cell_feature_matrix or .h5 file cell_feature_matrix.h5:
Xenium_unzipped
· | — cell_feature_matrix.h5
· | — cell_feature_matrix
· · | - barcodes.tsv.gz
· · | - features.tsv.gz
· · | - matrix.mtx.gz
· | — cells.parquet

Optional files to add to the metadata() as a list of paths (will be converted to parquet): · | — transcripts.parquet
· | — cell_boundaries.parquet
· | — nucleus_boundaries.parquet
· | — experiment.xenium
See addParquetPathsXenium()

Value

a `SpatialExperiment` or a `SingleCellExperiment` object

Author(s)

Yixing Estella Dong

Examples

```
# A relatively small data set is the Xenium mouse brain data that can be
```

```
# downloaded from 10X website.

# A mock .h5 and mock metadata with spatial location generated for a 4 genes by
# 6 cells object is in /extdata:

xepath <- system.file(
  file.path("extdata", "Xenium_small"),
  package = "SpatialExperimentIO")

list.files(xepath)

# One of the following depending on your input (.h5 or folder) and output
# (`SPE` or `SCE`) requirement.
xe_spe <- readXeniumSXE(dirName = xepath)
## Not run:
xe_spe <- readXeniumSXE(dirName = xepath, countMatPattern = "cell_feature_matrix")

## End(Not run)
xe_sce <- readXeniumSXE(dirName = xepath, returnType = "SCE")

xe_spe <- readXeniumSXE(dirName = xepath, addParquetPaths = TRUE)
xe_spe <- readXeniumSXE(dirName = xepath, addParquetPaths = TRUE, addNucBound = FALSE)
```

Index

.sanityCheck, 2

addParquetPathsCosmx, 3, *8, 9*
addParquetPathsXenium, 4, *15*
addParquetPathToMeta, 6

csvToParquetPaths, 7

readCosmxSXE, 7
readMerscopeSXE, 9
readSeqfishSXE, 11
readStarmapplusSXE, 12
readXeniumSXE, 14

SingleCellExperiment, *9, 10*, *12, 13*, *15*
SpatialExperiment, *7*, *9–15*