

# Package ‘spacexr’

April 24, 2025

**Type** Package

**Title** SpatialeXpressionR: Cell Type Identification in Spatial Transcriptomics

**Version** 1.0.0

**Description** Spatial-eXpression-R (spacexr) is a package for analyzing cell types in spatial transcriptomics data. This implementation is a fork of the spacexr GitHub repo (<https://github.com/dmcable/spacexr>), adapted to work with Bioconductor objects. The original package implements two statistical methods: RCTD for learning cell types and CSIDE for inferring cell type-specific differential expression. Currently, this fork only implements RCTD, which learns cell type profiles from annotated RNA sequencing (RNA-seq) reference data and uses these profiles to identify cell types in spatial transcriptomic pixels while accounting for platform-specific effects. Future releases will include an implementation of CSIDE.

**URL** <https://github.com/ggrajeda/spacexr>

**BugReports** <https://github.com/ggrajeda/spacexr/issues>

**Depends** R (>= 4.5.0)

**License** GPL (>= 3)

**Encoding** UTF-8

**Imports** ggplot2, Matrix, parallel, quadprog, httr, methods, memoise, BiocParallel, BiocFileCache, SummarizedExperiment, scatterpie, SpatialExperiment

**RoxygenNote** 7.3.2

**VignetteBuilder** knitr

**Suggests** BiocStyle, knitr, rmarkdown, testthat

**Config/testthat/edition** 3

**biocViews** GeneExpression, DifferentialExpression, SingleCell, RNASeq, Software, Spatial, Transcriptomics

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

**git\_branch** RELEASE\_3\_21

**git\_last\_commit** a8a9650

**git\_last\_commit\_date** 2025-04-15

**Repository** Bioconductor 3.21

**Date/Publication** 2025-04-23

**Author** Dylan Cable [aut],

Gabriel Grajeda [cre] (ORCID: <<https://orcid.org/0009-0003-7242-7476>>),

Fannie and John Hertz Foundation [fnd]

**Maintainer** Gabriel Grajeda <[gabriel.grajeda@gmail.com](mailto:gabriel.grajeda@gmail.com)>

## Contents

spacexr-package . . . . .	3
cache_Q_all . . . . .	4
cell_types . . . . .	4
cell_types<- . . . . .	5
cell_type_info . . . . .	5
cell_type_info<- . . . . .	6
chooseSigmaC . . . . .	6
computeCellTypeInfo . . . . .	7
config . . . . .	8
config<- . . . . .	8
coords . . . . .	9
coords<- . . . . .	9
counts . . . . .	10
counts<- . . . . .	10
createCellTypeInfo . . . . .	11
createRctd . . . . .	11
createRctdConfig . . . . .	14
createReference . . . . .	16
createSpatialRNA . . . . .	17
create_spe_doublet . . . . .	18
create_spe_from_columns . . . . .	18
create_spe_full . . . . .	19
create_spe_multi . . . . .	20
filterPixelsAndGetVars . . . . .	20
fitBulk . . . . .	21
fitPixels . . . . .	22
getDeGenes . . . . .	23
getNormRef . . . . .	24
internal_vars . . . . .	25
internal_vars<- . . . . .	25
load_Q_all . . . . .	26
load_SQ_all . . . . .	26
make_cache . . . . .	26
nUMI . . . . .	27
nUMI<- . . . . .	27

plotAllWeights . . . . .	28
plotCellTypeWeight . . . . .	29
process_beads_batch . . . . .	30
RctdConfig-class . . . . .	31
rctdSim . . . . .	32
Reference-class . . . . .	33
referenceToCellTypeInfo . . . . .	34
runRctd . . . . .	34
set_likelihood_vars . . . . .	36
spatialRNA . . . . .	37
SpatialRNA-class . . . . .	38
spatialRNA<- . . . . .	38
summarizedExperimentToSpatialRNA . . . . .	39
url_ok . . . . .	40

**Index****41**


---

spacexr-package	<i>spacexr: Cell Type Identification in Spatial Transcriptomics</i>
-----------------	---

---

**Description**

Spatial-eXpression-R (spacexr) is a package for analyzing cell types in spatial transcriptomics data. This implementation is a fork of the spacexr GitHub repo (<https://github.com/dmcable/spacexr>), adapted to work with Bioconductor objects. The original package implements two statistical methods: RCTD for learning cell types and CSIDE for inferring cell type-specific differential expression. Currently, this fork only implements RCTD, which learns cell type profiles from annotated RNA sequencing (RNA-seq) reference data and uses these profiles to identify cell types in spatial transcriptomic pixels while accounting for platform-specific effects. Future releases will include an implementation of CSIDE.

**Running RCTD**

To get started, create a `SpatialExperiment` object (called `spatial` here) for the spatial transcriptomics data and a `SummarizedExperiment` object (called `reference` here) for the RNA-seq data. Then simply run RCTD as:

```
rctd_data <- createRctd(spatial, reference)
results <- runRctd(rctd_data)
```

**Author(s)**

**Maintainer:** Gabriel Grajeda <[gabriel.grajeda@gmail.com](mailto:gabriel.grajeda@gmail.com)> ([ORCID](#))

Authors:

- Dylan Cable <[dmcable@umich.edu](mailto:dmcable@umich.edu)>

Other contributors:

- Fannie and John Hertz Foundation [funder]

**See Also**

Useful links:

- <https://github.com/ggrajeda/spacexr>
- Report bugs at <https://github.com/ggrajeda/spacexr/issues>

---

cache\_Q\_all

*Caches Q matrices*

---

**Description**

Stores Q matrices in cache after retrieving them from the Bioconductor Open Storage Network

**Usage**

```
cache_Q_all()
```

**Value**

list of bfcadd results

---

cell\_types

*Generic accessor for cell\_types slot*

---

**Description**

Generic accessor for cell\_types slot

**Usage**

```
cell_types(object)
```

```
## S4 method for signature 'Reference'
cell_types(object)
```

```
## S4 replacement method for signature 'Reference'
cell_types(object) <- value
```

**Arguments**

object            An object with a cell\_types slot

**Value**

The cell\_types slot of the object

---

cell\_types<-                    *Generic setter for cell\_types slot*

---

**Description**

Generic setter for cell\_types slot

**Usage**

```
cell_types(object) <- value
```

**Arguments**

object	An object with a cell_types slot
value	The new value for the cell_types slot

**Value**

The updated object

---

cell\_type\_info                *Generic accessor for cell\_type\_info slot*

---

**Description**

Generic accessor for cell\_type\_info slot

**Usage**

```
cell_type_info(object)

## S4 method for signature 'RctdConfig'
cell_type_info(object)

## S4 replacement method for signature 'RctdConfig'
cell_type_info(object) <- value
```

**Arguments**

object	An object with a cell_type_info slot
--------	--------------------------------------

**Value**

The cell\_type\_info slot of the object

---

cell\_type\_info<-            *Generic setter for cell\_type\_info slot*

---

**Description**

Generic setter for cell\_type\_info slot

**Usage**

```
cell_type_info(object) <- value
```

**Arguments**

object            An object with a cell\_type\_info slot  
value            The new value for the cell\_type\_info slot

**Value**

The updated object

---

chooseSigmaC            *Estimates sigma\_c by maximum likelihood*

---

**Description**

Estimates sigma\_c by maximum likelihood

**Usage**

```
chooseSigmaC(RCTD)
```

**Arguments**

RCTD            an [RctdConfig](#) object after running the [fitBulk](#) function.

**Value**

Returns an [RctdConfig](#) with the estimated sigma\_c.

## Examples

```
data(rctdSim)

# Spatial transcriptomics data
library(SpatialExperiment)
spatial_spe <- SpatialExperiment(
  assay = rctdSim$spatial_rna_counts,
  spatialCoords = rctdSim$spatial_rna_coords
)

# Reference data
library(SummarizedExperiment)
reference_se <- SummarizedExperiment(
  assays = list(counts = rctdSim$reference_counts),
  colData = rctdSim$reference_cell_types
)

# Create RCTD configuration
rctd_data <- createRctd(spatial_spe, reference_se)
rctd <- createRctdConfig(rctd_data)
rctd <- fitBulk(rctd)
rctd <- chooseSigmaC(rctd)
results <- fitPixels(rctd, rctd_mode = "doublet")
```

---

computeCellTypeInfo    *Computes cell type profiles in a scRNA-seq dataset*

---

## Description

Computes averaged normalized expression (summing to 1) for all cells within a cell type

## Usage

```
computeCellTypeInfo(raw.data, cell_types, nUMI, cell_type_names = NULL)
```

## Arguments

raw.data	a Digital Gene Expression matrix, with gene names as rownames and single cells as columns (barcodes for colnames)
cell_types	a named list of cell type assignment for each cell in raw.data
nUMI	a named list of total UMI count for each cell in raw.data
cell_type_names	a list of cell type names to compute profiles for. If NULL, uses the levels of cell_types

**Value**

Returns `cell_type_info`, a list of three elements: (1) `cell_type_means` (a `data_frame` (genes by cell types) for mean normalized expression) (2) `cell_type_names` (a list of cell type names) and (3) the number of cell types

---

<code>config</code>	<i>Generic accessor for config slot</i>
---------------------	---

---

**Description**

Generic accessor for config slot

**Usage**

```
config(object)

## S4 method for signature 'RctdConfig'
config(object)

## S4 replacement method for signature 'RctdConfig'
config(object) <- value
```

**Arguments**

`object`            An object with a config slot

**Value**

The config slot of the object

---

<code>config&lt;-</code>	<i>Generic setter for config slot</i>
--------------------------	---------------------------------------

---

**Description**

Generic setter for config slot

**Usage**

```
config(object) <- value
```

**Arguments**

`object`            An object with a config slot  
`value`             The new value for the config slot

**Value**

The updated object

---

coords	<i>Generic accessor for coords slot</i>
--------	---

---

**Description**

Generic accessor for coords slot

**Usage**

```
coords(object)

## S4 method for signature 'SpatialRNA'
coords(object)

## S4 replacement method for signature 'SpatialRNA'
coords(object) <- value
```

**Arguments**

object            An object with a coords slot

**Value**

The coords slot of the object

---

coords<-	<i>Generic setter for coords slot</i>
----------	---------------------------------------

---

**Description**

Generic setter for coords slot

**Usage**

```
coords(object) <- value
```

**Arguments**

object            An object with a coords slot  
value             The new value for the coords slot

**Value**

The updated object

---

counts	<i>Generic accessor for counts slot</i>
--------	---

---

**Description**

Generic accessor for counts slot

**Usage**

```
counts(object)

## S4 method for signature 'SpatialRNA'
counts(object)

## S4 replacement method for signature 'SpatialRNA'
counts(object) <- value

## S4 method for signature 'Reference'
counts(object)

## S4 replacement method for signature 'Reference'
counts(object) <- value
```

**Arguments**

object            An object with a counts slot

**Value**

The counts slot of the object

---

counts<-	<i>Generic setter for counts slot</i>
----------	---------------------------------------

---

**Description**

Generic setter for counts slot

**Usage**

```
counts(object) <- value
```

**Arguments**

object            An object with a counts slot  
value             The new value for the counts slot

**Value**

The updated object

---

createCellTypeInfo	<i>Create cell type information</i>
--------------------	-------------------------------------

---

**Description**

Create cell type information

**Usage**

```
createCellTypeInfo(
  reference = NULL,
  cell_type_names = NULL,
  cell_type_profiles = NULL,
  ref_n_cells_min = 25
)
```

**Arguments**

reference	<a href="#">Reference</a> object or NULL if using cell_type_profiles
cell_type_names	character vector of cell type names to include, optional
cell_type_profiles	matrix of precomputed cell type expression profiles (genes by cell type), optional
ref_n_cells_min	numeric, minimum number of cells per cell type in the reference (default: 25)

**Value**

A list containing cell type information

---

createRctd	<i>Preprocess data before RCTD</i>
------------	------------------------------------

---

**Description**

Performs initial preprocessing steps on a spatial transcriptomics dataset and a reference dataset prior to running RCTD. This function filters pixels and genes based on UMI counts and other thresholds, and identifies differentially expressed genes. The output of this function should be passed to runRctd to perform the cell type deconvolution.

**Usage**

```

createRctd(
  spatial_experiment,
  reference_experiment,
  cell_type_col = "cell_type",
  require_int = TRUE,
  gene_cutoff = 0.000125,
  fc_cutoff = 0.5,
  gene_cutoff_reg = 2e-04,
  fc_cutoff_reg = 0.75,
  gene_obs_min = 3,
  pixel_count_min = 10,
  UMI_min = 100,
  UMI_max = 2e+07,
  UMI_min_sigma = 300,
  ref_UMI_min = 100,
  ref_n_cells_min = 25,
  ref_n_cells_max = 10000,
  cell_type_profiles = NULL,
  class_df = NULL,
  cell_type_names = NULL
)

```

**Arguments**

`spatial_experiment`

[SummarizedExperiment](#) object (or any derivative object, including [SpatialExperiment](#)) containing spatial transcriptomics data to be deconvolved. The object must contain:

- An assay matrix of gene expression counts (genes as rows, pixels as columns) with unique gene names as row names and unique pixel barcodes as column names.
- Optionally, a `spatialCoords` matrix containing x and y coordinates for each pixel. If `spatial_experiment` does not have `spatialCoords`, dummy coordinates will be used.
- Optionally, a `colData` column named `nUMI` containing total UMI counts for each pixel. If not provided, `nUMI` will be calculated as the column sums of the counts matrix.

`reference_experiment`

[SummarizedExperiment](#) object containing annotated RNA-seq data (e.g., from snRNA-seq, scRNA-seq, or cell type-specific bulk RNA-seq), used to learn cell type profiles. The object must contain:

- An assay matrix of gene expression counts (genes as rows, cells as columns) with unique gene names as row names and unique cell barcodes as column names.
- A `colData` column containing cell type annotations for each cell (column name specified by `cell_type_col`).

- Optionally, a colData column named nUMI containing total UMI counts for each cell. If not provided, nUMI will be calculated as the column sums of the counts matrix.

cell_type_col	character, name of the entry in colData(reference_experiment) containing cell type annotations (default: "cell_type")
require_int	logical, whether counts and nUMI are required to be integers (default: TRUE)
gene_cutoff	numeric, minimum normalized gene expression for genes to be included in the platform effect normalization step (default: 0.000125)
fc_cutoff	numeric, minimum log fold change (across cell types) for genes to be included in the platform effect normalization step (default: 0.5)
gene_cutoff_reg	numeric, minimum normalized gene expression for genes to be included in the RCTD step (default: 0.0002)
fc_cutoff_reg	numeric, minimum log fold change (across cell types) for genes to be included in the RCTD step (default: 0.75)
gene_obs_min	numeric, minimum number of times a gene must appear in the spatial transcriptomics data to be included in the analysis (default: 3)
pixel_count_min	numeric, minimum total gene count for a pixel to be included in the analysis (default: 10)
UMI_min	numeric, minimum UMI count per pixel (default: 100)
UMI_max	numeric, maximum UMI count per pixel (default: 20,000,000)
UMI_min_sigma	numeric, minimum UMI count for pixels used in platform effect normalization (default: 300)
ref_UMI_min	numeric, minimum UMI count for cells to be included in the reference (default: 100)
ref_n_cells_min	numeric, minimum number of cells per cell type in the reference (default: 25)
ref_n_cells_max	numeric, maximum number of cells per cell type in the reference. Will down-sample if this number is exceeded. (default: 10,000)
cell_type_profiles	matrix of precomputed cell type expression profiles (genes by cell type), optional. If this option is used, gene names and cell type names must be present in the dimnames, and the reference will be ignored.
class_df	data frame mapping cell types to classes, optional. If specified, RCTD will report confidence on the class level.
cell_type_names	character vector of cell type names to include, optional

**Value**

A list with four elements:

- `spatial_experiment`: Preprocessed `SummarizedExperiment` object containing spatial transcriptomics data with filtered pixels and genes
- `cell_type_info`: List containing cell type information, including expression profiles and metadata
- `internal_vars`: List of internal variables used by RCTD, including differentially expressed gene lists and class information
- `config`: List of configuration parameters used for RCTD

### Examples

```
data(rctdSim)

# Spatial transcriptomics data
library(SpatialExperiment)
spatial_spe <- SpatialExperiment(
  assay = rctdSim$spatial_rna_counts,
  spatialCoords = rctdSim$spatial_rna_coords
)

# Reference data
library(SummarizedExperiment)
reference_se <- SummarizedExperiment(
  assays = list(counts = rctdSim$reference_counts),
  colData = rctdSim$reference_cell_types
)

# Filter spatial transcriptomics data and aggregate reference data
rctd_data <- createRctd(spatial_spe, reference_se)

# Run RCTD on filtered data
results <- runRctd(rctd_data, rctd_mode = "doublet", max_cores = 1)

# Access the cell type proportions (cell types as rows, pixels as columns)
assay(results, "weights")

# Check spot classifications for doublet mode
colData(results)$spot_class

# Access spatial coordinates
head(spatialCoords(results))
```

---

```
createRctdConfig
```

```
Create RCTD configuration object
```

---

### Description

Used internally by `runRctd`.

**Usage**

```
createRctdConfig(  
  rctd_data,  
  max_cores = 1,  
  max_multi_types = 4,  
  confidence_threshold = 5,  
  doublet_threshold = 20  
)
```

**Arguments**

rctd_data	list containing <code>createRctd</code> output
max_cores	numeric, maximum number of cores to use for parallel processing (default: 4)
max_multi_types	numeric, maximum number of cell types per pixel in multi mode (default: 4)
confidence_threshold	numeric, minimum change in likelihood (compared to other cell types) necessary to determine a cell type identity with confidence (default: 5)
doublet_threshold	numeric, penalty weight of predicting a doublet instead of a singlet for a pixel (default: 20)

**Details**

Default value of `max_cores` is set to 1 so that `devtools::check()` does not complain about parallelism in examples.

**Value**

RCTD configuration

**Examples**

```
data(rctdSim)  
  
# Spatial transcriptomics data  
library(SpatialExperiment)  
spatial_spe <- SpatialExperiment(  
  assay = rctdSim$spatial_rna_counts,  
  spatialCoords = rctdSim$spatial_rna_coords  
)  
  
# Reference data  
library(SummarizedExperiment)  
reference_se <- SummarizedExperiment(  
  assays = list(counts = rctdSim$reference_counts),  
  colData = rctdSim$reference_cell_types  
)
```

```
# Create RCTD configuration
rctd_data <- createRctd(spatial_spe, reference_se)
rctd <- createRctdConfig(rctd_data)
rctd <- fitBulk(rctd)
rctd <- chooseSigmaC(rctd)
results <- fitPixels(rctd, rctd_mode = "doublet")
```

---

createReference      [Reference](#) object constructor

---

## Description

[Reference](#) object constructor

## Usage

```
createReference(  
  counts,  
  cell_types,  
  nUMI = NULL,  
  require_int = TRUE,  
  n_max_cells = 10000,  
  min_UMI = 100  
)
```

## Arguments

counts	matrix (or dgCMatrix) of gene expression counts from RNA-seq data, with genes as rows and cells as columns (named by cell barcode)
cell_types	factor vector containing cell type annotations for each cell in the reference (identified by barcode). The factor levels represent the possible cell types.
nUMI	optional, numeric vector of total UMI counts per cell (identified by barcode). If not provided, nUMI will be calculated as the column sums of the counts matrix.
require_int	logical, whether counts and nUMI are required to be integers (default: TRUE)
n_max_cells	numeric, maximum number of cells per cell type. Will downsample if this number is exceeded. (default: 10,000)
min_UMI	numeric, minimum UMI count for cells to be included in the reference (default: 100)

## Value

[Reference](#) object

## Examples

```
data(rctdSim)

cell_types <- rctdSim$reference_cell_types[["cell_type"]]
names(cell_types) <- rownames(rctdSim$reference_cell_types)
reference <- createReference(rctdSim$reference_counts, cell_types)
```

---

createSpatialRNA      [SpatialRNA](#) object constructor

---

## Description

[SpatialRNA](#) object constructor

## Usage

```
createSpatialRNA(
  coords,
  counts,
  nUMI = NULL,
  use_fake_coords = FALSE,
  require_int = TRUE
)
```

## Arguments

coords	data frame (or matrix) containing x and y coordinates for each pixel (identified by barcode)
counts	matrix (or dgCMatrix) of gene expression counts, with genes as rows and pixels as columns (named by pixel barcode)
nUMI	optional, numeric vector of total UMI counts per pixel (identified by barcode). If not provided, nUMI will be calculated as the column sums of the counts matrix.
use_fake_coords	logical, whether the 'coords' parameter should be ignored and replaced with a placeholder coords matrix (default: FALSE)
require_int	logical, whether counts and nUMI are required to be integers (default: TRUE)

## Value

[SpatialRNA](#) object

**Examples**

```

data(rctdSim)

spatial_rna <- createSpatialRNA(
  as.data.frame(rctdSim$spatial_rna_coords),
  rctdSim$spatial_rna_counts
)

```

---

create_spe_doublet	<i>Converts the results of process_beads_batch to a SpatialExperiment</i>
--------------------	---

---

**Description**

Converts the results of process\_beads\_batch to a SpatialExperiment

**Usage**

```
create_spe_doublet(RCTD, results)
```

**Arguments**

RCTD	RctdConfig object
results	process_beads_batch results

**Value**

SpatialExperiment containing RCTD results

---

create_spe_from_columns	<i>Converts a list of RCTD results to a SpatialExperiment</i>
-------------------------	---

---

**Description**

The SpatialExperiment contains an assay with the cell type weights. Additional information (e.g., classification confidence) is stored in the colData, which contains the entries in the results list specified by the \*\_cols arguments. Spatial coordinates are stored in the spatialCoords.

**Usage**

```

create_spe_from_columns(
  RCTD,
  results,
  weights_col = "all_weights",
  character_cols = c(),
  logical_cols = c(),
  numeric_cols = c(),
  list_cols = c()
)

```

**Arguments**

RCTD	RctdConfig object
results	List of results (with named entries) for each pixel
weights_col	Name of list entry containing the cell type weights
character_cols	Names of list entries containing a character(1)
logical_cols	Names of list entries containing a logical(1)
numeric_cols	Names of list entries containing a numeric(1)
list_cols	Names of list entries containing a list

**Value**

SpatialExperiment containing RCTD results

---

create_spe_full	<i>Converts the results of decompose_batch to a SpatialExperiment</i>
-----------------	---

---

**Description**

Converts the results of decompose\_batch to a SpatialExperiment

**Usage**

```
create_spe_full(RCTD, results)
```

**Arguments**

RCTD	RctdConfig object
results	decompose_batch results

**Value**

SpatialExperiment containing RCTD results

---

create_spe_multi	<i>Converts the results of process_beads_multi to a SpatialExperiment</i>
------------------	---

---

**Description**

Converts the results of process\_beads\_multi to a SpatialExperiment

**Usage**

```
create_spe_multi(RCTD, results)
```

**Arguments**

RCTD	RctdConfig object
results	process_beads_multi results

**Value**

SpatialExperiment containing RCTD results

---

filterPixelsAndGetVars	<i>Filter pixels and create internal variables</i>
------------------------	--

---

**Description**

Filter pixels and create internal variables

**Usage**

```
filterPixelsAndGetVars(
  spatial_experiment,
  spatial_counts,
  cell_type_info,
  gene_cutoff = 0.000125,
  fc_cutoff = 0.5,
  gene_cutoff_reg = 2e-04,
  fc_cutoff_reg = 0.75,
  gene_obs_min = 3,
  pixel_count_min = 10,
  UMI_min = 100,
  UMI_max = 2e+07,
  UMI_min_sigma = 300,
  class_df = NULL
)
```

**Arguments**

spatial_experiment	<a href="#">SummarizedExperiment</a> object with spatial transcriptomics data
spatial_counts	spatial transcriptomics count matrix
cell_type_info	list containing cell type information
gene_cutoff	numeric, minimum normalized gene expression for genes to be included in the platform effect normalization step (default: 0.000125)
fc_cutoff	numeric, minimum log fold change (across cell types) for genes to be included in the platform effect normalization step (default: 0.5)
gene_cutoff_reg	numeric, minimum normalized gene expression for genes to be included in the RCTD step (default: 0.0002)
fc_cutoff_reg	numeric, minimum log fold change (across cell types) for genes to be included in the RCTD step (default: 0.75)
gene_obs_min	numeric, minimum number of times a gene must appear in the spatial transcriptomics data to be included in the analysis (default: 3)
pixel_count_min	numeric, minimum total gene count for a pixel to be included in the analysis (default: 10)
UMI_min	numeric, minimum UMI count per pixel (default: 100)
UMI_max	numeric, maximum UMI count per pixel (default: 20,000,000)
UMI_min_sigma	numeric, minimum UMI count for pixels used in platform effect normalization (default: 300)
class_df	data frame mapping cell types to classes, optional. If specified, RCTD will report confidence on the class level.

**Value**

List containing the filtered pixels and internal variables

---

fitBulk

*Performs Platform Effect Normalization:*


---

**Description**

Estimates bulk cell type composition and uses this to estimate platform effects and normalize cell type proportions

**Usage**

```
fitBulk(RCTD)
```

**Arguments**

RCTD                    an `RctdConfig` object after running the `createRctd` function.

**Value**

Returns an `RctdConfig` object normalized for platform effects.

**Examples**

```
data(rctdSim)

# Spatial transcriptomics data
library(SpatialExperiment)
spatial_spe <- SpatialExperiment(
  assay = rctdSim$spatial_rna_counts,
  spatialCoords = rctdSim$spatial_rna_coords
)

# Reference data
library(SummarizedExperiment)
reference_se <- SummarizedExperiment(
  assays = list(counts = rctdSim$reference_counts),
  colData = rctdSim$reference_cell_types
)

# Create RCTD configuration
rctd_data <- createRctd(spatial_spe, reference_se)
rctd <- createRctdConfig(rctd_data)
rctd <- fitBulk(rctd)
rctd <- chooseSigmaC(rctd)
results <- fitPixels(rctd, rctd_mode = "doublet")
```

---

fitPixels

*Runs the RCTD algorithm*


---

**Description**

If in doublet mode, fits at most two cell types per pixel. It classifies each pixel as 'singlet' or 'doublet' and searches for the cell types on the pixel. If in full mode, can fit any number of cell types on each pixel. In multi mode, cell types are added using a greedy algorithm, up to a fixed number.

**Usage**

```
fitPixels(RCTD, rctd_mode)
```

**Arguments**

**RCTD** an `RctdConfig` object after running the `chooseSigmaC` function.

**rctd\_mode** character string, either "doublet", "multi", or "full" on which mode to run RCTD. Please see above description.

**Value**

a `SpatialExperiment` object containing the results of the RCTD algorithm.

**Examples**

```
data(rctdSim)

# Spatial transcriptomics data
library(SpatialExperiment)
spatial_spe <- SpatialExperiment(
  assay = rctdSim$spatial_rna_counts,
  spatialCoords = rctdSim$spatial_rna_coords
)

# Reference data
library(SummarizedExperiment)
reference_se <- SummarizedExperiment(
  assays = list(counts = rctdSim$reference_counts),
  colData = rctdSim$reference_cell_types
)

# Create RCTD configuration
rctd_data <- createRctd(spatial_spe, reference_se)
rctd <- createRctdConfig(rctd_data)
rctd <- fitBulk(rctd)
rctd <- chooseSigmaC(rctd)
results <- fitPixels(rctd, rctd_mode = "doublet")
```

---

getDeGenes

*Returns a list of differentially expressed genes*

---

**Description**

For each cell type, chooses genes that have a minimum average normalized expression in that cell type, and whose expression is larger in that cell type than the average of all cell types. Filters out mitochondrial genes.

**Usage**

```

getDeGenes(
  spatial_counts,
  cell_type_info,
  fc_thresh = 1.25,
  expr_thresh = 0.00015,
  MIN_OBS = 3,
  de_type = "regression"
)

```

**Arguments**

`spatial_counts` spatial transcriptomics count matrix

`cell_type_info` cell type information and profiles of each cell, calculated from the scRNA-seq reference (see [computeCellTypeInfo](#))

`fc_thresh` minimum log<sub>e</sub> fold change required for a gene.

`expr_thresh` minimum expression threshold, as normalized expression (proportion out of 1, or counts per 1).

`MIN_OBS` the minimum number of occurrences of each gene in the SpatialRNA object.

`de_type` type of differential expression (i.e., "regression" or "bulk")

**Value**

a list of differentially expressed gene names

---

getNormRef	<i>Normalizes cell type profiles to a target dataset</i>
------------	--

---

**Description**

renormalizes `cell_type_means` to have average the same as the puck. The average for each gene is weighted by cell type proportions given by `proportions`.

**Usage**

```
getNormRef(puck, cell_type_means, gene_list, proportions)
```

**Arguments**

`puck` an object of type [SpatialRNA](#), the target dataset

`cell_type_means` a `data_frame` (genes by cell types) for mean normalized expression (see [computeCellTypeInfo](#))

`gene_list` a list of genes to be used for the normalization

`proportions` a named list (for each cell type) of proportion of the cell type on the bulk dataset (not constrained to sum to 1)

**Value**

Returns `cell_type_means`, a `data_frame` (genes by cell types) for mean normalized cell type expression profiles in which platform effects have been removed to match the [SpatialRNA](#) data.

---

<code>internal_vars</code>	<i>Generic accessor for internal_vars slot</i>
----------------------------	--

---

**Description**

Generic accessor for `internal_vars` slot

**Usage**

```
internal_vars(object)

## S4 method for signature 'RctdConfig'
internal_vars(object)

## S4 replacement method for signature 'RctdConfig'
internal_vars(object) <- value
```

**Arguments**

`object`            An object with an `internal_vars` slot

**Value**

The `internal_vars` slot of the object

---

<code>internal_vars&lt;-</code>	<i>Generic setter for internal_vars slot</i>
---------------------------------	--

---

**Description**

Generic setter for `internal_vars` slot

**Usage**

```
internal_vars(object) <- value
```

**Arguments**

`object`            An object with an `internal_vars` slot  
`value`             The new value for the `internal_vars` slot

**Value**

The updated object

---

load_Q_all	<i>Retrieves Q matrices from cache, populating the cache if necessary.</i>
------------	--

---

**Description**

Retrieves Q matrices from cache, populating the cache if necessary.

**Usage**

```
load_Q_all()
```

**Value**

list of matrices

---

load_SQ_all	<i>Retrieves SQ matrices from cache, populating the cache if necessary.</i>
-------------	---

---

**Description**

Retrieves SQ matrices from cache, populating the cache if necessary.

**Usage**

```
load_SQ_all()
```

**Value**

list of matrices

---

make_cache	<i>Returns a stateful function that stores the most recent non-null argument and returns it for NULL values.</i>
------------	--

---

**Description**

This is effectively used to manage global variables.

**Usage**

```
make_cache()
```

**Value**

cache function

---

nUMI	<i>Generic accessor for nUMI slot</i>
------	---------------------------------------

---

**Description**

Generic accessor for nUMI slot

**Usage**

```
nUMI(object)

## S4 method for signature 'SpatialRNA'
nUMI(object)

## S4 replacement method for signature 'SpatialRNA'
nUMI(object) <- value

## S4 method for signature 'Reference'
nUMI(object)

## S4 replacement method for signature 'Reference'
nUMI(object) <- value
```

**Arguments**

object	An object with a nUMI slot
--------	----------------------------

**Value**

The nUMI slot of the object

---

nUMI<-	<i>Generic setter for nUMI slot</i>
--------	-------------------------------------

---

**Description**

Generic setter for nUMI slot

**Usage**

```
nUMI(object) <- value
```

**Arguments**

object	An object with a nUMI slot
value	The new value for the nUMI slot

**Value**

The updated object

---

plotAllWeights	<i>Plot pie charts of cell type proportions across pixels</i>
----------------	---

---

**Description**

Generates a visualization where each pixel is represented by a pie chart showing the proportions of different cell types at that location. Users should run this function on the result of [runRctd](#).

**Usage**

```
plotAllWeights(  
  rctd_spe,  
  assay_name = "weights",  
  cell_type_colors = NA,  
  r = 0.4,  
  lwd = 1,  
  title = NA  
)
```

**Arguments**

rctd_spe	<a href="#">SpatialExperiment</a> object containing RCTD results
assay_name	character, name of the assay to plot (default: "weights")
cell_type_colors	vector of colors for the different cell types (default: rainbow)
r	numeric, radius of the pie charts (default: 0.4)
lwd	numeric, line width of the pie chart borders (default: 1)
title	character, plot title (default: NA)

**Details**

This function is adapted from `vizAllTopics` in the `STdeconvolve` package.

**Value**

ggplot object showing cell type proportions at each pixel using pie charts

## Examples

```
data(rctdSim)

# In practice, results_spe should contain the results of an RCTD run.
results_spe <- rctdSim$proportions_spe
plotAllWeights(
  results_spe, r = 0.05, lwd = 0.5, title = "Cell Type Proportions"
)
```

---

plotCellTypeWeight      *Plot pixel proportions for a specific cell type*

---

## Description

Creates a visualization showing how the proportion of a specific cell type varies across space, represented by point color intensity. Users should run this function on the result of [runRctd](#).

## Usage

```
plotCellTypeWeight(
  rctd_spe,
  cell_type,
  assay_name = "weights",
  size = 10,
  stroke = 1,
  alpha = 1,
  low = "white",
  high = "red",
  title = NA
)
```

## Arguments

rctd_spe	<a href="#">SpatialExperiment</a> object containing RCTD results
cell_type	character, name of cell type to plot
assay_name	character, name of the assay to plot (default: "weights")
size	numeric, size of the points (default: 10)
stroke	numeric, border width of the points (default: 1)
alpha	numeric, point transparency between 0 and 1 (default: 1)
low	color for the low end of the proportion color scale (default: "white")
high	color for the high end of the proportion color scale (default: "red")
title	character, plot title (default: NA)

**Details**

This function is adapted from vizTopic in the STdeconvolve package.

**Value**

ggplot object showing the proportion of a specified cell type at each pixel

**Examples**

```
data(rctdSim)

# In practice, results_spe should contain the results of an RCTD run.
results_spe <- rctdSim$proportions_spe
plotCellTypeWeight(
  results_spe, "ct1", size = 5, title = "Cell Type Density (ct1)"
)
```

---

process\_beads\_batch     *Runs RCTD in doublet mode on puck*

---

**Description**

Then, computes cell type proportions for each pixel in puck. Classifies each pixel as 'singlet' or 'doublet' and searches for the cell types on the pixel

**Usage**

```
process_beads_batch(
  cell_type_info,
  gene_list,
  puck,
  class_df = NULL,
  constrain = TRUE,
  MAX_CORES = 8,
  MIN.CHANGE = 0.001,
  confidence_threshold = 10,
  doublet_threshold = 25
)
```

**Arguments**

cell_type_info	cell type information and profiles of each cell, calculated from the scRNA-seq reference (see <a href="#">computeCellTypeInfo</a> )
gene_list	a list of genes to be used for RCTD
puck	an object of type <a href="#">SpatialRNA</a> , the target dataset
class_df	A dataframe mapping cell types to classes

constrain	logical whether to constrain the weights to sum to one on each pixel
MAX_CORES	number of cores to use (will use parallel processing if more than one).
MIN.CHANGE	(default 0.001) the minimum change in the norm of the WLS solution used to determine the cell type proportions
confidence_threshold	(Default 10) the minimum change in likelihood (compared to other cell types) necessary to determine a cell type identity with confidence
doublet_threshold	(Default 25) the penalty weight of predicting a doublet instead of a singlet for a pixel

**Value**

Returns results, a list of RCTD results for each pixel, which can be organized by feeding into [create\\_spe\\_doublet](#)

---

RctdConfig-class	<i>RCTD algorithm configuration</i>
------------------	-------------------------------------

---

**Description**

RCTD algorithm configuration

**Usage**

```
## S4 method for signature 'RctdConfig'
show(object)
```

**Arguments**

object            RCTD configuration object

**Slots**

spatialRNA a [SpatialRNA](#) object containing the processed spatial transcriptomics data for analysis

config a list of configuration options for the RCTD algorithm, set via [createRctd](#)

cell\_type\_info a named list containing cell type expression profiles with two components: info (raw profiles from reference data) and renorm (profiles normalized to match the spatial data)

internal\_vars a list of internal variables used during the RCTD analysis

---

`rctdSim`*Simulated spatial transcriptomics dataset*

---

### Description

A simulated dataset containing both reference single-cell RNA-seq data and spatial transcriptomics data. The dataset includes 750 genes across 3 cell types, with 50% of genes being differentially expressed between cell types. The spatial data consists of two kinds of cell type mixtures, documented below.

### Usage

```
data(rctdSim)
```

### Format

A list containing five components:

**reference\_counts** A matrix of simulated reference counts with 750 rows (genes) and 75 columns (25 samples per cell type). Gene names are of the form "g1", "g2", etc.

**reference\_cell\_types** A data frame specifying the cell type ("ct1", "ct2", "ct3") for each reference sample.

**spatial\_rna\_coords** A matrix with columns x and y giving the coordinates of each spatial transcriptomics pixel.

**spatial\_rna\_counts** A matrix of simulated spatial transcriptomics counts with 750 rows (genes) and 12 columns (spatial locations).

**proportions\_spe** A [SpatialExperiment](#) object containing the true cell type proportions for each spatial location. The weights assay contains a matrix with 3 rows (cell types) and 12 columns (spatial locations).

### Details

The dataset was generated using the following parameters:

- 750 genes, with 50% probability of differential expression
- 3 cell types with 25 reference samples each
- 12 spatial locations total:
  - 6 locations with mixture type 1 (90% ct1, 10% ct2)
  - 6 locations with mixture type 2 (20% ct1, 40% ct2, 40% ct3)

Base expression levels were sampled uniformly between 0 and 10. Differentially expressed genes were randomly selected to be either up-regulated (2x) or down-regulated (0.5x) in specific cell types. Final counts were generated using a Poisson distribution.

**Examples**

```

data(rctdSim)

# Spatial transcriptomics data
library(SpatialExperiment)
spatial_spe <- SpatialExperiment(
  assay = rctdSim$spatial_rna_counts,
  spatialCoords = rctdSim$spatial_rna_coords
)

# Reference data
library(SummarizedExperiment)
reference_se <- SummarizedExperiment(
  assays = list(counts = rctdSim$reference_counts),
  colData = rctdSim$reference_cell_types
)

# Access true cell type proportions
true_proportions <- assay(rctdSim$proportions_spe, "weights")

```

---

Reference-class	<i>RNA-seq reference data</i>
-----------------	-------------------------------

---

**Description**

A class representing annotated RNA sequencing data used as a reference to learn cell type profiles. The reference can come from single-nucleus RNA sequencing (snRNA-seq), single-cell RNA sequencing (scRNA-seq), or cell type-specific bulk RNA sequencing. RCTD uses these profiles to estimate cell type proportions in spatial transcriptomics data.

**Usage**

```

## S4 method for signature 'Reference'
show(object)

```

**Arguments**

```

object          Reference object

```

**Slots**

```

cell_types  factor vector containing cell type annotations for each cell in the reference (identified by barcode)
counts      sparse matrix of gene expression counts from RNA-seq data, with genes as rows and cells as columns (named by cell barcode)
nUMI        numeric vector of total UMI counts per cell (identified by barcode)

```

**Examples**

```
data(rctdSim)

cell_types <- rctdSim$reference_cell_types[["cell_type"]]
names(cell_types) <- rownames(rctdSim$reference_cell_types)
reference <- createReference(rctdSim$reference_counts, cell_types)
```

---

```
referenceToCellTypeInfo
```

*Process cell type information from a Reference object*

---

**Description**

Process cell type information from a Reference object

**Usage**

```
referenceToCellTypeInfo(reference, cell_type_names, CELL_MIN = 25)
```

**Arguments**

reference	Reference object
cell_type_names	character vector of cell type names to include
CELL_MIN	numeric, minimum number of cells per cell type in the reference (default: 25)

**Value**

List containing cell type information

---

```
runRctd
```

*Run RCTD algorithm to decompose cell type mixtures*

---

**Description**

Robust Cell Type Decomposition (RCTD) is a computational method for deconvolving cell type mixtures in spatial transcriptomics data. RCTD learns cell type profiles from annotated RNA sequencing (RNA-seq) reference data and uses these profiles to identify cell types in spatial transcriptomic pixels while accounting for platform-specific effects. The RCTD algorithm has three modes suited for different spatial technologies:

- **doublet**: Fits at most two cell types per pixel and classifies each pixel as a "singlet" (one cell type) or "doublet" (two cell types). Best for high spatial resolution technologies like Slide-seq or MERFISH, where pixels are more likely to contain only 1 or 2 cells.

- `multi`: Uses a greedy algorithm to fit up to `max_multi_types` cell types per pixel (default: 4). Best for lower resolution technologies like 100-micron Visium spots, which can contain more cell types.
- `full`: Fits any number of cell types per pixel without restrictions.

### Usage

```
runRctd(
  rctd_data,
  rctd_mode = c("doublet", "multi", "full"),
  max_cores = 4,
  max_multi_types = 4,
  confidence_threshold = 5,
  doublet_threshold = 20
)
```

### Arguments

<code>rctd_data</code>	list containing <a href="#">createRctd</a> output
<code>rctd_mode</code>	character string specifying the RCTD mode: "doublet", "multi", or "full" (default: "doublet")
<code>max_cores</code>	numeric, maximum number of cores to use for parallel processing (default: 4)
<code>max_multi_types</code>	numeric, maximum number of cell types per pixel in multi mode (default: 4)
<code>confidence_threshold</code>	numeric, minimum change in likelihood (compared to other cell types) necessary to determine a cell type identity with confidence (default: 5)
<code>doublet_threshold</code>	numeric, penalty weight of predicting a doublet instead of a singlet for a pixel (default: 20)

### Value

A [SpatialExperiment](#) object containing the RCTD results with:

- Three assays (one in full mode):
  - `weights`: Cell type proportions restricted according to the specified mode
  - `weights_unconfident`: Cell type proportions restricted according to the specified mode, including unconfident predictions (not available in full mode)
  - `weights_full`: Unrestricted cell type proportions (not available in full mode, use `weights` instead)

Assays have cell types as rows and pixels as columns, with values representing the proportion (0 to 1) of each cell type in each pixel. Assay columns sum to 1 (except for rejected pixels, which sum to 0).

- `spatialCoords` containing spatial coordinates for each pixel
- `colData` containing:

- For doublet mode:
  - \* spot\_class: Classification as "singlet", "doublet\_certain", "doublet\_uncertain", or "reject"
  - \* first\_type, second\_type: Predicted cell types
  - \* first\_class, second\_class: Whether predictions were made at the class level
  - \* Additional metrics like min\_score, singlet\_score
- For multi mode:
  - \* cell\_type\_list: List of cell types per pixel
  - \* conf\_list: List of whether cell type predictions are confident
  - \* Additional metrics like min\_score

## Examples

```

data(rctdSim)

# Spatial transcriptomics data
library(SpatialExperiment)
spatial_spe <- SpatialExperiment(
  assay = rctdSim$spatial_rna_counts,
  spatialCoords = rctdSim$spatial_rna_coords
)

# Reference data
library(SummarizedExperiment)
reference_se <- SummarizedExperiment(
  assays = list(counts = rctdSim$reference_counts),
  colData = rctdSim$reference_cell_types
)

# Filter spatial transcriptomics data and aggregate reference data
rctd_data <- createRctd(spatial_spe, reference_se)

# Run RCTD on filtered data
results <- runRctd(rctd_data, rctd_mode = "doublet", max_cores = 1)

# Access the cell type proportions (cell types as rows, pixels as columns)
assay(results, "weights")

# Check spot classifications for doublet mode
colData(results)$spot_class

# Access spatial coordinates
head(spatialCoords(results))

```

**Description**

Given a matrix, `Q_mat`, or  $\log P(y|x)$ , under the Poisson-Lognormal model. Sets this as a global variable for fast computations in the future.

**Usage**

```
set_likelihood_vars(Q_mat_loc, X_vals, sigma = NULL)
```

**Arguments**

<code>Q_mat_loc</code>	Matrix of precomputed probabilities, as previously computed by <code>{calc_Q_par}</code>
<code>X_vals</code>	the x-values used for computing the likelihood functions.
<code>sigma</code>	(default NULL). If NULL, computes <code>SQ_mat</code> according to <code>Q_mat_loc</code> . Else, uses precomputed values of <code>SQ_mat</code> stored in <code>SQ_mat_all</code> with index <code>sigma</code>

**Value**

Return value should be ignored.

---

spatialRNA

*Generic accessor for spatialRNA slot*


---

**Description**

Generic accessor for spatialRNA slot

**Usage**

```
spatialRNA(object)
```

```
## S4 method for signature 'RctdConfig'
```

```
spatialRNA(object)
```

```
## S4 replacement method for signature 'RctdConfig'
```

```
spatialRNA(object) <- value
```

**Arguments**

<code>object</code>	An object with a spatialRNA slot
---------------------	----------------------------------

**Value**

The spatialRNA slot of the object

---

SpatialRNA-class      *Spatial transcriptomics data*

---

### Description

A class representing spatial transcriptomics data, where gene expression is measured at fixed locations called "pixels" (also known as "spots" or "beads"). RCTD estimates the proportions of different cell types on each pixel.

### Usage

```
## S4 method for signature 'SpatialRNA'
show(object)
```

### Arguments

object                  SpatialRNA object

### Slots

coords data frame (or matrix) containing x and y coordinates for each pixel (identified by barcode)  
 counts sparse matrix of gene expression counts, with genes as rows and pixels as columns (named by pixel barcode)  
 nUMI numeric vector of total UMI counts per pixel (identified by barcode)

### Examples

```
data(rctdSim)

# Create SpatialRNA object
spatial_rna <- createSpatialRNA(
  as.data.frame(rctdSim$spatial_rna_coords),
  rctdSim$spatial_rna_counts
)
```

---

spatialRNA<-                  *Generic setter for spatialRNA slot*

---

### Description

Generic setter for spatialRNA slot

### Usage

```
spatialRNA(object) <- value
```

**Arguments**

object	An object with a spatialRNA slot
value	The new value for the spatialRNA slot

**Value**

The updated object

---

summarizedExperimentToSpatialRNA

*Convert a SummarizedExperiment to a SpatialRNA object*

---

**Description**

Convert a SummarizedExperiment to a SpatialRNA object

**Usage**

```
summarizedExperimentToSpatialRNA(spatial_experiment, require_int = TRUE)
```

**Arguments**

spatial\_experiment

[SummarizedExperiment](#) object (or any derivative object, including [SpatialExperiment](#)) containing spatial transcriptomics data to be deconvolved. The object must contain:

- An assay matrix of gene expression counts (genes as rows, pixels as columns) with unique gene names as row names and unique pixel barcodes as column names.
- Optionally, a spatialCoords matrix containing x and y coordinates for each pixel. If spatial\_experiment does not have spatialCoords, dummy coordinates will be used.
- Optionally, a colData column named nUMI containing total UMI counts for each pixel. If not provided, nUMI will be calculated as the column sums of the counts matrix.

require\_int      logical, whether counts and nUMI are required to be integers (default: TRUE)

**Value**

[SpatialRNA](#) object

---

url_ok	<i>Checks that a URL returns a 200 status code for a HEAD request</i>
--------	---

---

**Description**

Checks that a URL returns a 200 status code for a HEAD request

**Usage**

```
url_ok(url)
```

**Arguments**

url	character(1)
-----	--------------

**Value**

logical(1)

# Index

- \* **datasets**
  - rctdSim, 32
- \* **internal**
  - cache\_Q\_all, 4
  - cell\_type\_info, 5
  - cell\_type\_info<-, 6
  - cell\_types, 4
  - cell\_types<-, 5
  - chooseSigmaC, 6
  - computeCellTypeInfo, 7
  - config, 8
  - config<-, 8
  - coords, 9
  - coords<-, 9
  - counts, 10
  - counts<-, 10
  - create\_spe\_doublet, 18
  - create\_spe\_from\_columns, 18
  - create\_spe\_full, 19
  - create\_spe\_multi, 20
  - createCellTypeInfo, 11
  - createRctdConfig, 14
  - createReference, 16
  - createSpatialRNA, 17
  - filterPixelsAndGetVars, 20
  - fitBulk, 21
  - fitPixels, 22
  - getDeGenes, 23
  - getNormRef, 24
  - internal\_vars, 25
  - internal\_vars<-, 25
  - load\_Q\_all, 26
  - load\_SQ\_all, 26
  - make\_cache, 26
  - nUMI, 27
  - nUMI<-, 27
  - process\_beads\_batch, 30
  - RctdConfig-class, 31
  - Reference-class, 33
  - referenceToCellTypeInfo, 34
  - set\_likelihood\_vars, 36
  - spatialRNA, 37
  - SpatialRNA-class, 38
  - spatialRNA<-, 38
  - summarizedExperimentToSpatialRNA, 39
  - url\_ok, 40
- cache\_Q\_all, 4
- cell\_type\_info, 5
- cell\_type\_info, RctdConfig-method (cell\_type\_info), 5
- cell\_type\_info<-, 6
- cell\_type\_info<-, RctdConfig-method (cell\_type\_info), 5
- cell\_types, 4
- cell\_types, Reference-method (cell\_types), 4
- cell\_types<-, 5
- cell\_types<-, Reference-method (cell\_types), 4
- chooseSigmaC, 6, 23
- computeCellTypeInfo, 7, 24, 30
- config, 8
- config, RctdConfig-method (config), 8
- config<-, 8
- config<-, RctdConfig-method (config), 8
- coords, 9
- coords, SpatialRNA-method (coords), 9
- coords<-, 9
- coords<-, SpatialRNA-method (coords), 9
- counts, 10
- counts, Reference-method (counts), 10
- counts, SpatialRNA-method (counts), 10
- counts<-, 10
- counts<-, Reference-method (counts), 10
- counts<-, SpatialRNA-method (counts), 10
- create\_spe\_doublet, 18, 31
- create\_spe\_from\_columns, 18

create\_spe\_full, 19  
create\_spe\_multi, 20  
createCellTypeInfo, 11  
createRctd, 11, 15, 22, 31, 35  
createRctdConfig, 14  
createReference, 16  
createSpatialRNA, 17

filterPixelsAndGetVars, 20  
fitBulk, 6, 21  
fitPixels, 22

getDeGenes, 23  
getNormRef, 24

internal\_vars, 25  
internal\_vars,RctdConfig-method  
(internal\_vars), 25  
internal\_vars<-, 25  
internal\_vars<-,RctdConfig-method  
(internal\_vars), 25

load\_Q\_all, 26  
load\_SQ\_all, 26

make\_cache, 26

nUMI, 27  
nUMI,Reference-method (nUMI), 27  
nUMI,SpatialRNA-method (nUMI), 27  
nUMI<-, 27  
nUMI<-,Reference-method (nUMI), 27  
nUMI<-,SpatialRNA-method (nUMI), 27

plotAllWeights, 28  
plotCellTypeWeight, 29  
process\_beads\_batch, 30

RctdConfig, 6, 22, 23  
RctdConfig-class, 31  
rctdSim, 32  
Reference, 11, 16, 34  
Reference-class, 33  
referenceToCellTypeInfo, 34  
runRctd, 14, 28, 29, 34

set\_likelihood\_vars, 36  
show,RctdConfig-method  
(RctdConfig-class), 31  
show,Reference-method  
(Reference-class), 33

show,SpatialRNA-method  
(SpatialRNA-class), 38  
spacexr (spacexr-package), 3  
spacexr-package, 3  
SpatialExperiment, 3, 12, 28, 29, 32, 35, 39  
SpatialRNA, 17, 24, 25, 30, 31, 39  
spatialRNA, 37  
spatialRNA,RctdConfig-method  
(spatialRNA), 37  
SpatialRNA-class, 38  
spatialRNA<-, 38  
spatialRNA<-,RctdConfig-method  
(spatialRNA), 37  
SummarizedExperiment, 3, 12, 14, 21, 39  
summarizedExperimentToSpatialRNA, 39

url\_ok, 40