Package: spacexr (via r-universe)

May 30, 2026

Type Package

Title SpatialeXpressionR: Cell Type Identification in Spatial Transcriptomics

Version 1.5.0

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

Config/pak/sysreqs cmake libfontconfig1-dev libfreetype6-dev make libmagick++-dev gsfonts libicu-dev libuv1-dev libssl-dev zlib1g-dev

Repository https://bioc.r-universe.dev

Date/Publication 2026-04-28 13:04:48 UTC

RemoteUrl https://github.com/bioc/spacexr

RemoteRef HEAD

RemoteSha b981b934b5bb557afd4f60a38deacf80acfcfa9d

Contents

spacexr-package 2
createRctd 3
plotAllWeights 6
plotCellTypeWeight 7
rctdSim 8
runRctd 10

Index **13**

`spacexr-package` *spacexr: Cell Type Identification in Spatial Transcriptomics*

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> (ORCID)

Authors:

- Dylan Cable <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

`createRctd` *Preprocess data before RCTD*

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 downsample 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))
```

`plotAllWeights` *Plot pie charts of cell type proportions across pixels*

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`	`SpatialExperiment` 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	SpatialExperiment 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)"
)
```

---

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")
```

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

`rctd_data`	list containing `createRctd` output
`rctd_mode`	character string specifying the RCTD mode: `"doublet"`, `"multi"`, or `"full"` (default: `"doublet"`)
`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)

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))
```

Index

∗ **datasets**
rctdSim, 8

createRctd, 3, *10*

plotAllWeights, 6
plotCellTypeWeight, 7

rctdSim, 8
runRctd, *6, 7*, 10

spacexr *(*spacexr-package*)*, 2
spacexr-package, 2
SpatialExperiment, *2*, *4*, *6*, *8, 9*, *11*
SummarizedExperiment, *2*, *4, 5*