Package 'spacexr'

November 24, 2023

Type Package

Title SpatialeXpressionR: Cell type identification and cell type-specific differential expression in spatial transcriptomics

Version 2.2.1

Author Dylan Cable

Maintainer <dcable@mit.edu>

Description
cell type identification and cell type-specific differential expression for spatial transcriptomics

Depends R (>= 3.5.0)

License GNU General Public License v3.0.

Encoding UTF-8

LazyData true

Imports readr,
ggplot2,
pals,
Matrix,
parallel,
doParallel,
foreach,
quadprog,
tibble,
dplyr,
reshape2,
knitr,
rmarkdown,
fields,
mgcv,
CompQuadForm,
Rfast,
locfdr,
metafor

RoxygenNote 7.2.3

VignetteBuilder knitr

R topics documented:

`aggregate_cell_types` *Aggregates the pixel occurrences for each cell type in the* `RCTD` *object*

Description

The difference with `count_cell_types` is that this function does not filter out pixels based on total cell type weight, as occurs in the CSIDE algorithm.

Usage

```
aggregate_cell_types(myRCTD, barcodes, doublet_mode = T)
```

Arguments

`barcodes`	the barcodes, or pixel names, of the `SpatialRNA` object to be used when counting cell types.
`doublet_mode`	(default TRUE) if TRUE, uses RCTD doublet mode weights. Otherwise, uses RCTD full mode weights
`RCTD`	an `RCTD` object with annotated cell types e.g. from the `run.RCTD` function.

Value

a named vector of number of pixel occurrences for each cell type

---

`build.designmatrix.intercept`

Constructs a design matrix for running CSIDE with only an intercept term

---

Description

Constructs a design matrix for running CSIDE with only an intercept term

Usage

```
build.designmatrix.intercept(myRCTD, barcodes = NULL)
```

Arguments

`myRCTD`	an `RCTD` object with annotated cell types e.g. from the `run.RCTD` function.
`barcodes`	(default NULL) the barcodes, or pixel names, of the `SpatialRNA` object to be used when creating the design matrix.

Value

A design matrix containing the covariates for running CSIDE. The rownames represent pixel names and are a subset of the pixels in the `SpatialRNA` object. The column represents the intercept

---

`build.designmatrix.nonparam`

Constructs a design matrix for running CSIDE nonparametrically

---

Description

The design matrix contains thin plate spline basis functions spanning the space of smooth functions.

Usage

```
build.designmatrix.nonparam(myRCTD, barcodes = NULL, df = 15)
```

Arguments

`myRCTD`	an `RCTD` object with annotated cell types e.g. from the `run.RCTD` function.
`barcodes`	(default NULL) the barcodes, or pixel names, of the `SpatialRNA` object to be used when creating the design matrix.
`df`	(default 15) the degrees of freedom, or number of basis functions to be used in the design matrix

Value

A design matrix containing the covariates for running CSIDE. The rownames represent pixel names and are a subset of the pixels in the `SpatialRNA` object. The columns each represent a covariate for explaining differential expression.

build.designmatrix.regions

Constructs a design matrix for running CSIDE across a set of regions

Description

The design matrix contains for each region a column of 0s and 1s representing membership in that region.

Usage

```
build.designmatrix.regions(myRCTD, region_list)
```

Arguments

myRCTD	an RCTD object with annotated cell types e.g. from the run.RCTD function.
region_list	a list of character vectors, where each vector contains pixel names, or barcodes, for a single region. These pixel names should be a subset of the pixels in the SpatialRNA object

Value

A design matrix containing the covariates for running CSIDE. The rownames represent pixel names and are a subset of the pixels in the SpatialRNA object. The columns each represent a covariate for explaining differential expression.

build.designmatrix.single

Constructs a design matrix for running CSIDE with a single explanatory variable

Description

The design matrix contains an intercept column and a column of the explanatory variable.

Usage

```
build.designmatrix.single(myRCTD, explanatory.variable)
```

Arguments

myRCTD	an RCTD object with annotated cell types e.g. from the run.RCTD function.
explanatory.variable	a named numeric vector representing the explanatory variable used for explaining differential expression in CSIDE. Names of the variable are the SpatialRNA pixel names, and values should be standardized between 0 and 1.

Value

A design matrix containing the covariates for running CSIDE. The rownames represent pixel names and are a subset of the pixels in the SpatialRNA object. The columns each represent a covariate for explaining differential expression.

---

choose_sigma_c *Estimates sigma_c by maximum likelihood*

Description

Estimates sigma_c by maximum likelihood

Usage

```
choose_sigma_c(RCTD)
```

Arguments

RCTD	an RCTD object after running the fitBulk function.

Value

Returns an RCTD with the estimated sigma_c.

---

convert.old.RCTD *Updates an old RCTD object to be compatible with the current version of spacexr.*

Description

Updates an old RCTD object to be compatible with the current version of spacexr.

Usage

```
convert.old.RCTD(myRCTD)
```

Arguments

RCTD	an RCTD object (potentially from an older version.

Value

an RCTD object updated to be compatible with the current version of spacexr.

`count_cell_types` *Counts number of pixel occurrences for each cell type to be used in the CSIDE model*

Description

The difference with `aggregate_cell_types` is that this function does filter out pixels based on total cell type weight, as occurs in the CSIDE algorithm.

Usage

```
count_cell_types(
  myRCTD,
  barcodes,
  cell_types,
  cell_type_threshold = 125,
  doublet_mode = T,
  weight_threshold = NULL
)
```

Arguments

`barcodes`	the barcodes, or pixel names, of the `SpatialRNA` object to be used when counting cell typel\.
`cell_types`	the cell types used for CSIDE. If null, cell types will be chosen with aggregate occurences of at least 'cell_type_threshold', as aggregated by `aggregate_cell_types`
`cell_type_threshold`	(default 125) min occurence of number of cells for each cell type to be used, as aggregated by `aggregate_cell_types`
`doublet_mode`	(default TRUE) if TRUE, uses RCTD doublet mode weights. Otherwise, uses RCTD full mode weights
`weight_threshold`	(default NULL) the threshold of total normalized weights across all cell types in `cell_types` per pixel to be included in the model. Default 0.99 for doublet_mode or 0.95 for full_mode.
`RCTD`	an `RCTD` object with annotated cell types e.g. from the `run.RCTD` function.

Value

a named vector of number of pixel occurrences for each cell type

`create.RCTD` *Creates an* `RCTD` *object from a scRNA-seq reference* `Reference` *object and a* `SpatialRNA` *object*

Description

Creates an `RCTD` object from a scRNA-seq reference `Reference` object and a `SpatialRNA` object

Usage

```
create.RCTD(
  spatialRNA,
  reference,
  max_cores = 4,
  test_mode = FALSE,
  gene_cutoff = 0.000125,
  fc_cutoff = 0.5,
  gene_cutoff_reg = 2e-04,
  fc_cutoff_reg = 0.75,
  UMI_min = 100,
  UMI_max = 2e+07,
  counts_MIN = 10,
  UMI_min_sigma = 300,
  class_df = NULL,
  CELL_MIN_INSTANCE = 25,
  cell_type_names = NULL,
  MAX_MULTI_TYPES = 4,
  keep_reference = F,
  cell_type_profiles = NULL,
  CONFIDENCE_THRESHOLD = 5,
  DOUBLET_THRESHOLD = 20
)
```

Arguments

`spatialRNA`	a `SpatialRNA` object to run RCTD on
`reference`	a `Reference` object scRNA-seq reference used for RCTD
`max_cores`	for parallel processing, the number of cores used. If set to 1, parallel processing is not used. The system will additionally be checked for number of available cores.
`gene_cutoff`	minimum normalized gene expression for genes to be included in the platform effect normalization step.
`fc_cutoff`	minimum log-fold-change (across cell types) for genes to be included in the platform effect normalization step.
`gene_cutoff_reg`	minimum normalized gene expression for genes to be included in the RCTD step.
`fc_cutoff_reg`	minimum log-fold-change (across cell types) for genes to be included in the RCTD step.

UMI_min	minimum UMI per pixel included in the analysis
UMI_max	maximum UMI per pixel included in the analysis
counts_MIN	(default 10) minimum total counts per pixel of genes used in the analysis.
UMI_min_sigma	minimum UMI per pixel for the choose_sigma_c function
class_df	(optional) if not NULL, then a dataframe mapping each cell type to a cell class, so that RCTD will report confidence on the class level.
CELL_MIN_INSTANCE	minimum number of cells required per cell type. Default 25, can be lowered if desired.
cell_type_names	A list of cell types to be included from the reference. If NULL, uses all cell types
MAX_MULTI_TYPES	(multi-mode only) Default 4, max number of cell types per pixel
keep_reference	(Default FALSE) if true, keeps the reference object stored within the RCTD object
cell_type_profiles	Default NULL, option to pass in cell type profiles in directly as a genes by cell type matrix, including gene names and cell type names. If this option is used, reference will be ignored.
CONFIDENCE_THRESHOLD	(Default 5) the minimum change in likelihood (compared to other cell types) necessary to determine a cell type identity with confidence
DOUBLET_THRESHOLD	(Default 20) the penalty weight of predicting a doublet instead of a singlet for a pixel

Value

an RCTD object, which is ready to run the run.RCTD function

---

create.RCTD.replicates

Creates an RCTD.replicates object across multiple SpatialRNA replicates

---

Description

Applies the create.RCTD function for each SpatialRNA replicate inputted using a scRNA-seq reference Reference object.

Usage

```
create.RCTD.replicates(
  spatialRNA.replicates,
  reference,
  replicate_names,
  group_ids = NULL,
  max_cores = 4,
```

```
  test_mode = FALSE,
  gene_cutoff = 0.000125,
  fc_cutoff = 0.5,
  gene_cutoff_reg = 2e-04,
  fc_cutoff_reg = 0.75,
  UMI_min = 100,
  UMI_max = 2e+07,
  UMI_min_sigma = 300,
  class_df = NULL,
  CELL_MIN_INSTANCE = 25,
  cell_type_names = NULL,
  MAX_MULTI_TYPES = 4,
  keep_reference = F,
  CONFIDENCE_THRESHOLD = 5,
  DOUBLET_THRESHOLD = 20
)
```

Arguments

spatialRNA.replicates	a list of multiple SpatialRNA objects to run RCTD on
reference	a Reference object scRNA-seq reference used for RCTD
replicate_names	a character vector of names for each replicate provided in spatialRNA.replicates
group_ids	(default constant across replicates) a named integer vector (length number of replicates) containing the group id for each replicate. Names represent the replicate names, and replicates of the same group will be expected to be more similar than replicates across groups
max_cores	for parallel processing, the number of cores used. If set to 1, parallel processing is not used. The system will additionally be checked for number of available cores.
gene_cutoff	minimum normalized gene expression for genes to be included in the platform effect normalization step.
fc_cutoff	minimum log-fold-change (across cell types) for genes to be included in the platform effect normalization step.
gene_cutoff_reg	minimum normalized gene expression for genes to be included in the RCTD step.
fc_cutoff_reg	minimum log-fold-change (across cell types) for genes to be included in the RCTD step.
UMI_min	minimum UMI per pixel included in the analysis
UMI_max	maximum UMI per pixel included in the analysis
UMI_min_sigma	minimum UMI per pixel for the choose_sigma_c function
class_df	(optional) if not NULL, then a dataframe mapping each cell type to a cell class, so that RCTD will report confidence on the class level.
CELL_MIN_INSTANCE	minimum number of cells required per cell type. Default 25, can be lowered if desired.

`cell_type_names`
: A list of cell types to be included from the reference. If NULL, uses all cell types

`MAX_MULTI_TYPES`
: (multi-mode only) Default 4, max number of cell types per pixel

`keep_reference`
: (Default FALSE) if true, keeps the `reference` object stored within the `RCTD` object

`CONFIDENCE_THRESHOLD`
: (Default 5) the minimum change in likelihood (compared to other cell types) necessary to determine a cell type identity with confidence

`DOUBLET_THRESHOLD`
: (Default 20) the penalty weight of predicting a doublet instead of a singlet for a pixel

`cell_type_info`
: Default NULL, option to pass in `cell_type_info` directly

Value

an `RCTD.replicates` object, which is ready to run the `run.RCTD.replicates` function

---

create_RCTD_plots *Create all plots for an RCTD object after cell types have been assigned*

Description

Create all plots for an RCTD object after cell types have been assigned

Usage

```
create_RCTD_plots(myRCTD, datadir)
```

Arguments

`myRCTD`
: a `RCTD` object with annotated cell types e.g. from the `run.RCTD` function.

`datadir`
: directory where plots should be saved

CSIDE.population.inference *Runs population-level differential expression inference for a `RCTD.replicates` object*

Description

First, CSIDE must have been run on all replicates using e.g. the `run.CSIDE.replicates` function.

Usage

```
CSIDE.population.inference(
  RCTD.replicates,
  params_to_test = NULL,
  use.groups = FALSE,
  MIN.CONV.REPLICATES = 2,
  MIN.CONV.GROUPS = 2,
  CT.PROP = 0.5,
  fdr = 0.01,
  log_fc_thresh = 0.4,
  normalize_expr = F,
  meta = FALSE,
  meta.design.matrix = NULL,
  meta.test_var = "intrcpt"
)
```

Arguments

RCTD.replicates	a RCTD.replicates object for which to perform population-level DE inference. Note, at least three replicates must be provided.
use.groups	(default FALSE) if TRUE, treats the replicates as having multiple groups (e.g. samples) according to the group_ids slot
MIN.CONV.REPLICATES	(default 2) the minimum number of replicates (if not use.groups) for which a gene must converge
MIN.CONV.GROUPS	(default 2) the minimum number of groups (if use.groups) for which a gene must converge
CT.PROP	(default 0.5) minimum ratio of gene expression within cell type compared to other cell types
fdr	(default 0.01) false discovery rate
log_fc_thresh	(default 0.4) minimum natural log estimated DE threshold
normalize_expr	(default FALSE) if TRUE, constrains total gene expression to sum to 1 in each condition
meta	(default FALSE) if TRUE, conducts population inference using general meta regression
meta.design.matrix	(default NULL) if meta == TRUE, then this is the design matrix for meta regression. Rows are samples and named columns are covariates.
meta.test_var	(default 'intrcpt') if meta == TRUE, this indicates which variable is tested in the meta regression. By default the intercept is tested, but one can also test for fixed effects of other covariates.
params_to_test:	(default 2 for test_mode = 'individual', all parameters for test_mode = 'categorical'). An integer vector of parameter indices to test. Note, for population mode, only the first parameter is tested.

Value

an RCTD.replicates object containing the results of the CSIDE population-level algorithm. See RCTD.replicates for documentation on the population_de_results, population_sig_gene_list, and population_sig_gene_df objects.

---

exvar.celltocell.interactions *Constructs an explanatory variable representing density of a cell type*

Description

This explanatory variable can be used with CSIDE to detect cell-to-cell interactions. Density is computing using an exponentially-decaying filter. Currently only works for doublet mode RCTD.

Usage

```
exvar.celltocell.interactions(myRCTD, barcodes, cell_type, radius = 50)
```

Arguments

myRCTD	an RCTD object with annotated cell types e.g. from the run.RCTD function.
barcodes	the barcodes, or pixel names, of the SpatialRNA for which to evaluate the explanatory variable. These would be the pixels used in the C-SIDE model.
cell_type	the cell type (character) for which to compute density.
radius	(default 50) the radius of the exponential filter. Approximately, the distance considered to be a relevant interaction.

Value

explanatory.variable a named numeric vector representing the explanatory variable used for explaining differential expression in CSIDE. Names of the variable are the SpatialRNA pixel names, and values are standardized between 0 and 1. This variable represents density of the selected cell type.

---

exvar.point.density *Constructs an explanatory variable representing density of a set of points*

Description

This explanatory variable can be used with CSIDE to detect DE in the proximity of these points. Density is computing using an exponentially-decaying filter.

Usage

```
exvar.point.density(myRCTD, barcodes, points, radius = 50)
```

Arguments

myRCTD	an RCTD object with annotated cell types e.g. from the run.RCTD function.
barcodes	the barcodes, or pixel names, of the SpatialRNA for which to evaluate the explanatory variable. These would be the pixels used in the C-SIDE model.
points	a N by 2 matrix containing the locations of the points to be used for computing density. The first column should be the x coordinates while the second column should be the y coordinate.
radius	(default 50) the radius of the exponential filter. Approximately, the distance considered to be a relevant interaction.

Value

explanatory.variable a named numeric vector representing the explanatory variable used for explaining differential expression in CSIDE. Names of the variable are the SpatialRNA pixel names, and values are standardized between 0 and 1. This variable represents density of the given point set.

---

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 RCTD object after running the create.RCTD function.

Value

Returns an RCTD object normalized for platform effects.

---

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, doublet_mode = "doublet")
```

Arguments

RCTD	an RCTD object after running the choose_sigma_c function.
doublet_mode	character string, either "doublet", "multi", or "full" on which mode to run RCTD. Please see above description.

Value

an RCTD object containing the results of the RCTD algorithm.

---

get_cell_type_info *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

```
get_cell_type_info(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

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

---

get_decomposed_data *Decomposes SpatialRNA data into individual cells*

Description

Warning: in the current RCTD version, this function is deprecated, and is no longer supported. For differential expression tasks, we instead recommend the RCTDE method.

Usage

```
get_decomposed_data(
  results_df,
  gene_list,
  puck,
  weights_doublet,
  cell_type_info
)
```

Arguments

results_df	a dataframe of RCTD results
gene_list	a list of genes to be used for the decomposition
puck	an object of type SpatialRNA
weights_doublet	a dataframe of predicted weights in doublet mode
cell_type_info	cell type information and profiles of each cell, calculated from the scRNA-seq reference (see get_cell_type_info)

Details

Applied to the output of gather_results. Singlet pixels are left unchanged, and doublet_certain conditions are decomposed into single cells.

Value

An object of type SpatialRNA representing the decomposed cells

---

get_de_genes *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

```
get_de_genes(
  cell_type_info,
  puck,
  fc_thresh = 1.25,
  expr_thresh = 0.00015,
  MIN_OBS = 3
)
```

Arguments

cell_type_info	cell type information and profiles of each cell, calculated from the scRNA-seq reference (see get_cell_type_info)
puck	an object of type SpatialRNA
fc_thresh	minimum log_e 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 occurances of each gene in the SpatialRNA object.

Value

a list of differntially expressed gene names

get_doublet_weights — *Converts RCTD doublet mode results to a weight matrix (across all cell types)*

Description

RCTD must have been run in doublet mode

Usage

```
get_doublet_weights(myRCTD)
```

Arguments

RCTD	an RCTD object with annotated cell types from the run.RCTD function.

Value

a weights matrix of cell type proportions for each pixel and each cell type.

get_norm_ref — *Normalizes cell type profiles to a target dataset*

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

```
get_norm_ref(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 get_cell_type_info)
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.

get_standard_errors *On an RCTD object after running CSIDE, returns an array of standard errors of CSIDE coefficients*

Description

The dimensions of the standard error array is N_genes x N_coefficients x N_cell_types The N_coefficients are the number of explanatory variables in the CSIDE model

Usage

```
get_standard_errors(myRCTD)
```

Arguments

myRCTD	an RCTD object with fitted CSIDE e.g. from the run.CSIDE function.

Value

a three-dimensional array representing CSIDE standard errors for each gene, each coefficient, and each cell type.

import_weights *Assigns a cell type 'weights' matrix to an RCTD object*

Description

Assigns a cell type 'weights' matrix to an RCTD object

Usage

```
import_weights(myRCTD, weights)
```

Arguments

myRCTD	a RCTD object to be assigned weights.
weights	a matrix of weights (pixels by cell types). weights must be normalized to have rows sum to 1. Furthermore, rownames and colnames must be assigned as pixel names and cell types respectively.

Value

the RCTD object with weights assigned.

make_all_de_plots *Makes all CSIDE plots on RCTD object, after running CSIDE*

Description

Makes all CSIDE plots on RCTD object, after running CSIDE

Usage

```
make_all_de_plots(myRCTD, datadir)
```

Arguments

myRCTD	RCTD object containing de_results, after running CSIDE
datadir	output directory

make_de_plots_genes *Makes spatial gene CSIDE plots (colored continuously) on RCTD object, after running CSIDE*

Description

Makes spatial gene CSIDE plots (colored continuously) on RCTD object, after running CSIDE

Usage

```
make_de_plots_genes(myRCTD, datadir)
```

Arguments

myRCTD	RCTD object containing de_results, after running CSIDE
datadir	output directory

make_de_plots_quant *Makes quantitative CSIDE plots on RCTD object, after running CSIDE*

Description

Makes quantitative CSIDE plots on RCTD object, after running CSIDE

Usage

```
make_de_plots_quant(myRCTD, datadir)
```

Arguments

myRCTD	RCTD object containing de_results, after running CSIDE
datadir	output directory

make_de_plots_regions *Makes spatial gene CSIDE plots (colored by discrete regions) on RCTD object, after running CSIDE*

Description

Makes spatial gene CSIDE plots (colored by discrete regions) on RCTD object, after running CSIDE

Usage

```
make_de_plots_regions(myRCTD, datadir)
```

Arguments

myRCTD	RCTD object containing de_results, after running CSIDE
datadir	output directory

make_de_plots_replicates *Makes spatial gene CSIDE plots (colored by two discrete regions) on RCTD replicates object, after running CSIDE*

Description

Runs on genes that were identified as significant at the population level

Usage

```
make_de_plots_replicates(RCTD.replicates, datadir)
```

Arguments

RCTD.replicates	a RCTD.replicates object after performing population-level DE inference.
datadir	output directory

Details

These plots are colored by two discrete regions based on high or low explanatory variable values. Bold points represent expressed, whereas unbold points represent pixels not expressing the gene.

make_de_plots_spatial *Makes spatial gene CSIDE plots (colored by two discrete regions) on RCTD object, after running CSIDE*

Description

These plots are colored by two discrete regions based on high or low explanatory variable values. Bold points represent expressed, whereas unbold points represent pixels not expressing the gene.

Usage

```
make_de_plots_spatial(myRCTD, datadir)
```

Arguments

myRCTD	RCTD object containing de_results, after running CSIDE
datadir	output directory

merge_RCTD_objects *Creates an RCTD.replicates object across multiple RCTD objects*

Description

Creates an RCTD.replicates object across multiple RCTD objects

Usage

```
merge_RCTD_objects(RCTD.reps, replicate_names, group_ids = NULL)
```

Arguments

RCTD.reps	a list of multiple RCTD objects to merge into one RCTD.replicates object.
replicate_names	a character vector of names for each replicate provided in RCTD.reps
group_ids	(default constant across replicates) a named integer vector (length number of replicates) containing the group id for each replicate. Names represent the replicate names, and replicates of the same group will be expected to be more similar than replicates across groups.

Value

an RCTD.replicates object, containing each RCTD object in RCTD.reps

normalize_weights *Normalizes the 'weights' matrix from the RCTD results object*

Description

Normalizes the 'weights' matrix from the RCTD results object

Usage

```
normalize_weights(weights)
```

Arguments

weights	a matrix of weights to be normalized

Value

norm.weights a normalized matrix of weights where rows sum to one.

plot_all_cell_types *Plots all cell types in space*

Description

Plots the first cell type in doublet mode. Saves as 'all_cell_types.pdf'

Usage

```
plot_all_cell_types(results_df, coords, cell_type_names, resultsdir)
```

Arguments

results_df	a dataframe of RCTD results (see gather_results)
coords	a dataframe of coordinates of each pixel
cell_type_names	list of cell type names
resultsdir	output directory

Value

returns ggplot2 object

plot_class — *Plots a factor variable in space on the puck*

Description

Colors points based on class

Usage

```
plot_class(puck, barcodes_cur, my_class, counter_barcodes = NULL, title = NULL)
```

Arguments

puck	an object of type SpatialRNA
barcodes_cur	a list of barcodes to include in the plot
my_class	a named (by barcode) factor list for the coloring

Value

Returns a ggplot object

plot_cond_occur — *Barplot of the confident counts for each cell type*

Description

Plots the number of confident labels in 'full_mode'. Saves as 'cell_type_occur.pdf'

Usage

```
plot_cond_occur(cell_type_names, resultsdir, weights, puck)
```

Arguments

cell_type_names	list of cell type names
resultsdir	output directory
weights	a dataframe of RCTD output weights (see gather_results)
puck	an object of type SpatialRNA

Value

returns ggplot2 object

plot_doublets *Plots all doublets in space*

Description

saves as 'all_doublets.pdf'

Usage

```
plot_doublets(puck, doublets, resultsdir, cell_type_names)
```

Arguments

puck	an object of type SpatialRNA
doublets	a dataframe of RCTD results restricted to doublets
resultsdir	output directory
cell_type_names	list of cell type names

Value

returns ggplot2 object

plot_doublets_type *Plots doublets of each cell type individually*

Description

Plots the first cell type in doublet mode. Saves as 'all_doublets_type.pdf'

Usage

```
plot_doublets_type(puck, doublets_base, resultsdir, cell_type_names)
```

Arguments

puck	an object of type SpatialRNA
doublets_base	a dataframe of RCTD results restricted to doublets
resultsdir	output directory
cell_type_names	list of cell type names

plot_doub_occur_stack *Plots doublet co-occurances*

Description

Plots the doublet co-occurances. Saves as 'doublet_stacked_bar.pdf'

Usage

```
plot_doub_occur_stack(doub_occur, resultsdir, cell_type_names)
```

Arguments

doub_occur	a table of occurances of doublets
resultsdir	output directory
cell_type_names	list of cell type names

Value

returns ggplot2 object

plot_gene_raw *Makes a spatial plot of continuous gene expression for a particular gene*

Description

Units counts per 500

Usage

```
plot_gene_raw(myRCTD, gene, cell_type, ymax = 10)
```

Arguments

myRCTD	RCTD object containing de_results, after running CSIDE
gene	gene to be plotted
cell_type	cell_type to be plotted (only single cell type pixels)
ymax	(default 10) maximum expression (in counts per 500) for color scale

Value

gene expression plot

`plot_gene_regions` *Makes a spatial plot of gene expression for a particular gene This plot is colored by several discrete regions based on a categorical design matrix. Bold points represent expressed, whereas unbold points represent pixels not expressing the gene.*

Description

Makes a spatial plot of gene expression for a particular gene This plot is colored by several discrete regions based on a categorical design matrix. Bold points represent expressed, whereas unbold points represent pixels not expressing the gene.

Usage

```
plot_gene_regions(
  myRCTD,
  cell_type,
  gene,
  pixel_weight_thresh = 0.8,
  expr_thresh = 0
)
```

Arguments

`myRCTD`	RCTD object containing `de_results`, after running CSIDE
`cell_type`	cell_type to be plotted (only single cell type pixels)
`gene`	gene to be plotted
`pixel_weight_thresh`	(default 0.8) minimum cell type weight for pixels that are included
`expr_thresh`	(default 0) the minimum expression threshold to clear to be considered to be expressed

Value

gene expression plot

`plot_gene_two_regions` *Makes a spatial plot of gene expression for a particular gene This plot is colored by two discrete regions based on high or low explanatory variable values. Bold points represent expressed, whereas unbold points represent pixels not expressing the gene.*

Description

Makes a spatial plot of gene expression for a particular gene This plot is colored by two discrete regions based on high or low explanatory variable values. Bold points represent expressed, whereas unbold points represent pixels not expressing the gene.

Usage

```
plot_gene_two_regions(
  myRCTD,
  gene,
  cell_type,
  min_UMI = 200,
  expr.thresh = 0,
  exvar_thresh = 0.5
)
```

Arguments

myRCTD	RCTD object containing de_results, after running CSIDE
gene	gene to be plotted
cell_type	cell_type to be plotted (only single cell type pixels)
min_UMI	(default 200) minimum UMI for pixels that are included
expr.thresh	(default 0) the minimum expression threshold to clear to be considered to be expressed
exvar_thresh	threshold of the explanatory variable in order for points to be sorted into the two regions

Value

gene expression plot

---

plot_occur_unthreshold

Barplot of the counts for each cell type

Description

Plots the number of (including unconfident) labels in 'full_mode'. Saves as 'cell_type_occur_unthreshold.pdf'

Usage

```
plot_occur_unthreshold(cell_type_info, resultsdir, weights)
```

Arguments

cell_type_info	cell type information and profiles (see get_cell_type_info)
resultsdir	output directory
weights	a dataframe of RCTD output weights (see gather_results)

plot_prediction_gene *Makes a spatial plot of CSIDE fitted gene expression*

Description

Units counts per 500

Usage

```
plot_prediction_gene(myRCTD, cell_type, gene)
```

Arguments

myRCTD	RCTD object containing de_results, after running CSIDE
cell_type	cell_type to be plotted (only single cell type pixels)
gene	gene to be plotted

Value

plot of fitted gene expression

plot_puck_continuous *Plots a continuous value over locations on the puck*

Description

Colors points based on value of the function

Usage

```
plot_puck_continuous(
  puck,
  barcodes,
  plot_val,
  ylimit = c(0, 1),
  title = NULL,
  counter_barcodes = NULL,
  label = F,
  my_pal = NULL,
  xlim = NULL,
  ylim = NULL,
  size = 0.15,
  alpha = 1,
  small_point = F
)
```

Arguments

puck	an object of type SpatialRNA
barcodes	a list of barcodes to include in the plot
plot_val	a named (by barcode) list of values to plot
ylimit	minimum and maximum values for the range of plot as a numeric list
xlim	(optional) minimum and maximum value for x coordinate as a numeric list
ylim	(optional) minimum and maximum value for y coordinate as a numeric list
size	numeric size of points
cell_type_info	cell type information and profiles (see get_cell_type_info)

Value

Returns a ggplot object

---

plot_puck_wrapper *Plots a continuous value over filtered locations on the puck*

Description

Colors points based on value of the function, filtered for e.g. UMI and cell type

Usage

```
plot_puck_wrapper(
  puck,
  plot_val,
  cell_type = NULL,
  minUMI = 0,
  maxUMI = 2e+05,
  min_val = NULL,
  max_val = NULL,
  title = NULL,
  my_cond = NULL
)
```

Arguments

puck	an object of type SpatialRNA
plot_val	a named (by barcode) list of values to plot
cell_type	string specifying cell type to plot. if NULL, then all cell types are plotted
minUMI	numeric, minimum value for total UMIs to filter pixels
maxUMI	numeric, maximum value for total UMIs to filter pixels
min_val	numeric, minimum value for the range of plot as a numeric list
max_val	numeric, maximum value for the range of plot as a numeric list
cell_type_info	cell type information and profiles (see get_cell_type_info)

Value

Returns a ggplot object

plot_weights *Spatially plot the confident weights for each cell type*

Description

Plots the confident weights for each cell type as in full_mode. Saves as 'cell_type_weights.pdf'

Usage

```
plot_weights(cell_type_names, puck, resultsdir, weights)
```

Arguments

cell_type_names	list of cell type names
puck	an object of type SpatialRNA
resultsdir	output directory
weights	a dataframe of RCTD output weights (see gather_results)

plot_weights_doublet *Spatially plot the weights for each cell type in doublet_mode*

Description

Plots the weights for each cell type as in doublet_mode. Saves as 'cell_type_weights_doublet.pdf'

Usage

```
plot_weights_doublet(
  cell_type_names,
  puck,
  resultsdir,
  weights_doublet,
  results_df
)
```

Arguments

cell_type_names	list of cell type names
puck	an object of type SpatialRNA
resultsdir	output directory
weights_doublet	a dataframe of RCTD output weights for doublets (see gather_results)
results_df	dataframe of RCTD results (see gather_results)

plot_weights_unthreshold *Spatially plot all weights for each cell type*

Description

Plots all weights for each cell type as in full_mode. Saves as 'cell_type_weights_unthreshold.pdf'

Usage

```
plot_weights_unthreshold(cell_type_names, puck, resultsdir, weights)
```

Arguments

cell_type_names	list of cell type names
puck	an object of type SpatialRNA
resultsdir	output directory
weights	a dataframe of RCTD output weights (see gather_results)

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 = T,
  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 get_cell_type_info)
gene_list	a list of genes to be used for RCTD
puck	an object of type SpatialRNA, the target dataset
class_df	A dataframe returned by get_class_df to map cell types to classes
constrain	logical whether to constrain the weights to sum to one on each pixel
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
max_cores	number of cores to use (will use parallel processing if more than one).

Value

Returns results, a list of RCTD results for each pixel, which can be organized by feeding into gather_results

---

process_data *Runs RCTD in full mode on* puck

Description

Renormalizes cell_type_means to have average the same as the puck if proportions is given. Then, computes cell type proportions for each pixel in puck.

Usage

```
process_data(
  puck,
  gene_list,
  cell_type_info,
  proportions = NULL,
  trust_model = FALSE,
  constrain = T,
  OLS = F
)
```

Arguments

puck	an object of type SpatialRNA, the target dataset
gene_list	a list of genes to be used for RCTD
cell_type_info	cell type information and profiles of each cell, calculated from the scRNA-seq reference (see get_cell_type_info)
proportions	(optional) If given, a named list (for each cell type) of proportion of the cell type on the bulk dataset (not constrained to sum to 1)
constrain	logical whether to constrain the weights to sum to one on each pixel

Value

Returns `test_results`, a list of three items: (1) `conf_mat` a confusion matrix (not relevant) (2) `weights` a dataframe of predicted weights (3) a named list of predicted cell types

---

`RCTD-class` *An S4 class used to run the RCTD and CSIDE algorithms*

Description

Created using the `create.RCTD` function, a user can run RCTD using the `run.RCTD` function.

Slots

`spatialRNA` a `SpatialRNA` object containing the Spatial RNA dataset to be used for RCTD

`originalSpatialRNA` a `SpatialRNA` object containing the Spatial RNA dataset with all genes

`reference` a `Reference` object containing the cell type-labeled single cell reference

`config` a list of configuration options, set using the `create.RCTD` function

`cell_type_info` a named list of cell type profiles (means), containing two elements: `info`, directly calculated from the scRNA-seq reference, and `renorm`, renormalized the match the SpatialRNA dataset.

`internal_vars` a list of internal variables used by RCTD's computation

`results` (created after running RCTD) a list of results_df (a dataframe of RCTD results in doublet mode), weights (a dataframe of RCTD predicted weights in full mode), and weights_doublet (a dataframe of predicted weights in doublet mode, with cell type information in results_df).

In doublet-mode, The results of 'doublet_mode' are stored in '@results$results_df' and '@results$weights_doublet', the weights of each cell type. More specifically, the 'results_df' object contains one column per pixel (barcodes as rownames). Important columns are: * 'spot_class', a factor variable representing RCTD's classification in doublet mode: "singlet" (1 cell type on pixel), "doublet_certain" (2 cell types on pixel), "doublet_uncertain" (2 cell types on pixel, but only confident of 1), "reject" (no prediction given for pixel). * Next, the 'first_type' column gives the first cell type predicted on the bead (for all spot_class conditions except "reject"). * The 'second_type column' gives the second cell type predicted on the bead for doublet spot_class conditions (not a confident prediction for "doublet_uncertain").

Note that in multi-mode, results consists of a list of results for each pixel, which contains all_weights (weights from full mode), cell_type_list (cell types on multi mode), conf_list (which cell types are confident on multi mode) and sub_weights (proportions of cell types on multi mode).

`de_results` results of the CSIDE algorithm. Contains 'gene_fits', which contains the results of fits on individual genes, whereas 'res_gene_list' is a list, for each cell type, of significant genes detected by CSIDE.

`internal_vars_de` a list of variables that are used internally by CSIDE

RCTD.replicates-class *An S4 class used to store multiple replicates as SpatialRNA objects.*

Description

By storing multiple SpatialRNA replicates in this one object, it is convenient to run RCTD and CSIDE across all replicates. Finally, multiple replicates can be combined with population-level differential expression inference using the CSIDE.population.inference function

Details

Created using the create.RCTD.replicates or merge_RCTD_objects functions. One can run RCTD using the run.RCTD.replicates function, and one can run CSIDE using the run.CSIDE.replicates function.

Slots

RCTD.reps a list of RCTD objects, one for each replicate

population_de_results A list, indexed by cell type, of dataframes summarizing population-level differential expression for each genes. Relevant columns include: tau, variance across replicates; log_fc_est, the estimated differential expresison; sd_est, the standard error of estimated DE

population_sig_gene_list A list, indexed by cell type, of vectors of significant genes

population_sig_gene_df A list, indexed by cell type, of dataframe summarizing population-level differential expression for each signifcant gene, similar to population_de_results. Additionally, contains p (representing p-values) and q_val (representing q-values).

groups_ids a named integer vector (length number of replicates) containing the group id for each replicate. Names represent the replicate names, and replicates of the same group will be expected to be more similar than replicates across groups

read.SpatialRNA *Creates a SpatialRNA object from a coords and counts file*

Description

Warning: this function is provided out of convenience for experienced users, but we can not provide direct support for debugging file input errors. If you are obtaining errors from this method, we recommend a less error-prone procedure of loading in your coords and counts matrices in first and then using the 'SpatialRNA' constructor function, which will systematically check for errors in the inputs.

Usage

```
read.SpatialRNA(datadir, count_file = "counts.csv", coords_file = "coords.csv")
```

Arguments

datadir	(character) full path to input directory
count_file	(character) file name of the counts csv file (genes by pixels matrix)
coord_file	(character) file name of the coords csv file (pixels by (barcodes, x, y) matrix)

Details

The coords matrix needs to be formated as columns (barcodes, x, y)

Value

Returns a SpatialRNA object containing the coordinates and counts from the input files

---

read.VisiumSpatialRNA *Creates a SpatialRNA object from a 10x Genomics Visium 'outs' directory*

Description

Given a SpatialRNA directory 10x Genomics Visium 'outs' directory and returns a SpatialRNA object.

Usage

```
read.VisiumSpatialRNA(datadir)
```

Arguments

datadir	(string) full path to the 10x Genomics Visium 'outs' directory

Value

Returns a SpatialRNA object containing the coordinates and counts from the input files

---

Reference *constructor of Reference object*

Description

constructor of Reference object

Usage

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

Arguments

counts	A matrix (or dgCmatrix) representing Digital Gene Expression (DGE). Rownames should be genes and colnames represent barcodes/cell names.
cell_types	A named (by cell barcode) factor of cell type for each cell. The 'levels' of the factor would be the possible cell type identities.
nUMI	Optional, a named (by cell barcode) list of total counts or UMI's appearing at each pixel. If not provided, nUMI will be assumed to be the total counts appearing on each pixel.
n_max_cells	(default 10,000) the maximum number of cells per cell type. Will downsample if this number is exceeded. Counts should be untransformed count-level data
min_UMI	(default 100) minimum UMI count for cells to be included in the reference.

Value

Returns a Reference object containing the counts matrix, cell type labels, and UMI vector from the input files

---

Reference-class *An S4 class to represent Single-Cell RNA-seq reference*

Description

An S4 class to represent Single-Cell RNA-seq reference

Slots

cell_types a factor of cell type identities for each cell

counts a sparse matrix of raw counts for each gene (rowname) and each cell (colnames or barcodes)

nUMI an atomic vector of numeric UMI counts per cell

---

restrict_counts *Restricts a SpatialRNA object to a subset of genes (and applies a UMI threshold)*

Description

Restricts a SpatialRNA object to a subset of genes (and applies a UMI threshold)

Usage

```
restrict_counts(
  puck,
  gene_list,
  UMI_thresh = 1,
  UMI_max = 20000,
  counts_thresh = 1
)
```

Arguments

puck	a SpatialRNA object
gene_list	a list of gene names
UMI_thresh	minimum UMI per pixel
UMI_max	maximum UMI per pixel
counts_thresh	minimum counts per pixel (for genes in gene_list)

Value

Returns a SpatialRNA with counts filtered based on UMI threshold and gene list

---

restrict_puck *Restricts a SpatialRNA object to a subset of pixels*

Description

Given a SpatialRNA object and a list of barcodes (pixels), will return a SpatialRNA object restricted to the barcodes.

Usage

```
restrict_puck(puck, barcodes)
```

Arguments

puck	a SpatialRNA object
barcodes	a list of barcode names, a subset of rownames(puck@coords)

Value

Returns a SpatialRNA object subsampled to the barcodes

---

run.CSIDE *Runs cell type specific CSIDE on a RCTD object with a general design matrix*

Description

Identifies cell type specific differential expression (DE) across a general design matrix of covariates. Uses maximum likelihood estimation to estimate DE and standard errors for each gene and each cell type. Selects genes with significant nonzero DE. The type of test is determined by test_mode, and the parameters tested is determined by params_to_test.

Usage

```
run.CSIDE(
  myRCTD,
  X,
  barcodes,
  cell_types = NULL,
  gene_threshold = 5e-05,
  cell_type_threshold = 125,
  doublet_mode = T,
  test_mode = "individual",
  weight_threshold = NULL,
  sigma_gene = T,
  PRECISION.THRESHOLD = 0.05,
  cell_types_present = NULL,
  test_genes_sig = T,
  fdr = 0.01,
  cell_type_specific = NULL,
  params_to_test = NULL,
  normalize_expr = F,
  logs = F,
  log_fc_thresh = 0.4,
  cell_type_filter = NULL,
  test_error = F,
  fdr_method = "BH"
)
```

Arguments

myRCTD	an RCTD object with annotated cell types e.g. from the run.RCTD function.
X	a matrix containing the covariates for running CSIDE. The rownames represent pixel names and should be a subset of the pixels in the SpatialRNA object. The columns each represent a covariate for explaining differential expression and need to be linearly independent.
barcodes	the barcodes, or pixel names, of the SpatialRNA object to be used when fitting the model.
cell_types	the cell types used for CSIDE. If null, cell types will be chosen with aggregate occurences of at least 'cell_type_threshold', as aggregated by aggregate_cell_types
gene_threshold	(default 5e-5) minimum average normalized expression required for selecting genes
cell_type_threshold	(default 125) min occurence of number of cells for each cell type to be used, as aggregated by aggregate_cell_types
doublet_mode	(default TRUE) if TRUE, uses RCTD doublet mode weights. Otherwise, uses RCTD full mode weights
test_mode	(default 'individual') if 'individual', tests for DE individually for each parameter. If 'categorical', then tests for differences across multiple categorical parameters
weight_threshold	(default NULL) the threshold of total normalized weights across all cell types in cell_types per pixel to be included in the model. Default 0.99 for doublet_mode or 0.8 for full_mode.

`sigma_gene` (default TRUE) if TRUE, fits gene specific overdispersion parameter. If FALSE, overdispersion parameter is same across all genes.

`PRECISION.THRESHOLD` (default 0.05) for checking for convergence, the maximum parameter change per algorithm step

`cell_types_present` cell types (a superset of 'cell_types') to be considered as occuring often enough to consider for gene expression contamination during the step filtering out marker genes of other cell types.

`test_genes_sig` (default TRUE) logical controlling whether genes will be tested for significance

`fdr` (default 0.01) false discovery rate for hypothesis testing

`normalize_expr` (default FALSE) if TRUE, constrains total gene expression to sum to 1 in each condition. Setting normalize_expr = TRUE is only valid for testing single parameters with test_mode = 'individual'.

`logs` (default FALSE) if TRUE, writes progress to logs/de_logs.txt

`log_fc_thresh` (default 0.4) the natural log fold change cutoff for differential expression

`test_error` (default FALSE) if TRUE, exits after testing for error messages without running CSIDE. If set to TRUE, this can be used to quickly evaluate if CSIDE will run without error.

`fdr_method` (default BH) if BH, uses the Benjamini-Hochberg method. Otherwise, uses local fdr with an empirical null.

`cell_type_specific:` (default TRUE for all covariates). A logical vector of length the number of covariates indicating whether each covariate's DE parameters should be cell type-specific or shared across all cell types.

`params_to_test:` (default 2 for test_mode = 'individual', all parameters for test_mode = 'categorical'). An integer vector of parameter indices to test. For example c(1,4,5) would test only parameters corresponding to columns 1, 4, and 5 of the design matrix.

Value

an RCTD object containing the results of the CSIDE algorithm. Contains objects `de_results`, which contain the results of the CSIDE algorithm including 'gene_fits', which contains the results of fits on individual genes, in addition 'sig_gene_list', a list, for each cell type, of significant genes detected by CSIDE. Additionally, the object contains 'internal_vars_de' a list of variables that are used internally by CSIDE

`run.CSIDE.general` *Runs CSIDE on a RCTD object with a general design matrix*

Description

Identifies differential expression (DE) across a general design matrix of covariates. DE parameters can be cell type-specific or shared across all cell types. Uses maximum likelihood estimation to estimate DE and standard errors for each gene and each cell type. Selects genes with significant nonzero DE. The type of test is determined by `test_mode`, and the parameters tested is determined by `params_to_test`.

Usage

```
run.CSIDE.general(
  myRCTD,
  X1,
  X2,
  barcodes,
  cell_types = NULL,
  gene_threshold = 5e-05,
  cell_type_threshold = 125,
  doublet_mode = T,
  test_mode = "individual",
  weight_threshold = NULL,
  sigma_gene = T,
  PRECISION.THRESHOLD = 0.05,
  cell_types_present = NULL,
  test_genes_sig = T,
  fdr = 0.01,
  params_to_test = NULL,
  normalize_expr = F,
  logs = F,
  cell_type_filter = NULL,
  log_fc_thresh = 0.4,
  test_error = FALSE,
  fdr_method = "BH"
)
```

Arguments

myRCTD	an RCTD object with annotated cell types e.g. from the run.RCTD function.
X1	a matrix containing the covariates shared across all cell types. The rownames represent pixel names and should be a subset of the pixels in the SpatialRNA object. The columns each represent a covariate for explaining differential expression and need to be linearly independent.
X2	a matrix containing the cell type-specific covariates. The rownames represent pixel names and should be a subset of the pixels in the SpatialRNA object. The columns each represent a covariate for explaining differential expression and need to be linearly independent.
barcodes	the barcodes, or pixel names, of the SpatialRNA object to be used when fitting the model.
cell_types	the cell types used for CSIDE. If null, cell types will be chosen with aggregate occurences of at least 'cell_type_threshold', as aggregated by aggregate_cell_types
gene_threshold	(default 5e-5) minimum average normalized expression required for selecting genes
cell_type_threshold	(default 125) min occurence of number of cells for each cell type to be used, as aggregated by aggregate_cell_types
doublet_mode	(default TRUE) if TRUE, uses RCTD doublet mode weights. Otherwise, uses RCTD full mode weights
test_mode	(default 'individual') if 'individual', tests for DE individually for each parameter. If 'categorical', then tests for differences across multiple categorical parameters

`weight_threshold`
: (default NULL) the threshold of total normalized weights across all cell types in `cell_types` per pixel to be included in the model. Default 0.99 for doublet_mode or 0.8 for full_mode.

`sigma_gene`
: (default TRUE) if TRUE, fits gene specific overdispersion parameter. If FALSE, overdispersion parameter is same across all genes.

`PRECISION.THRESHOLD`
: (default 0.05) for checking for convergence, the maximum parameter change per algorithm step

`cell_types_present`
: cell types (a superset of 'cell_types') to be considered as occuring often enough to consider for gene expression contamination during the step filtering out marker genes of other cell types.

`test_genes_sig`
: (default TRUE) logical controlling whether genes will be tested for significance

`fdr`
: (default 0.01) false discovery rate for hypothesis testing

`normalize_expr`
: (default FALSE) if TRUE, constrains total gene expression to sum to 1 in each condition. Setting normalize_expr = TRUE is only valid for testing single parameters with test_mode = 'individual'.

`logs`
: (default FALSE) if TRUE, writes progress to logs/de_logs.txt

`log_fc_thresh`
: (default 0.4) the natural log fold change cutoff for differential expression

`test_error`
: (default FALSE) if TRUE, exits after testing for error messages without running CSIDE. If set to TRUE, this can be used to quickly evaluate if CSIDE will run without error.

`fdr_method`
: (default BH) if BH, uses the Benjamini-Hochberg method. Otherwise, uses local fdr with an empirical null.

`params_to_test:`
: (default 2 for test_mode = 'individual', all parameters for test_mode = 'categorical'). An integer vector of parameter indices to test. For example c(1,4,5) would test only parameters corresponding to columns 1, 4, and 5 of the design matrix X2.

Value

an RCTD object containing the results of the CSIDE algorithm. Contains objects `de_results`, which contain the results of the CSIDE algorithm including 'gene_fits', which contains the results of fits on individual genes, in addition 'sig_gene_list', a list, for each cell type, of significant genes detected by CSIDE, whereas 'all_gene_list' is the analogous list for all genes (including nonsignificant). Additionally, the object contains 'internal_vars_de' a list of variables that are used internally by CSIDE

`run.CSIDE.intercept` *Runs CSIDE on a RCTD object with only an intercept term*

Description

Identifies cell type specific gene expression for each cell type.

Usage

```
run.CSIDE.intercept(
  myRCTD,
  barcodes = NULL,
  cell_types = NULL,
  cell_type_threshold = 125,
  gene_threshold = 5e-05,
  doublet_mode = T,
  weight_threshold = NULL,
  sigma_gene = T,
  PRECISION.THRESHOLD = 0.05,
  cell_types_present = NULL,
  normalize_expr = F,
  logs = F,
  test_error = F
)
```

Arguments

myRCTD	an RCTD object with annotated cell types e.g. from the run.RCTD function.
barcodes	(default NULL) the barcodes, or pixel names, of the SpatialRNA object to be used when creating the design matrix.
cell_types	the cell types used for CSIDE. If null, cell types will be chosen with aggregate occurrences of at least 'cell_type_threshold', as aggregated by aggregate_cell_types
cell_type_threshold	(default 125) min occurrence of number of cells for each cell type to be used, as aggregated by aggregate_cell_types
gene_threshold	(default 5e-5) minimum average normalized expression required for selecting genes
doublet_mode	(default TRUE) if TRUE, uses RCTD doublet mode weights. Otherwise, uses RCTD full mode weights
weight_threshold	(default NULL) the threshold of total normalized weights across all cell types in cell_types per pixel to be included in the model. Default 0.99 for doublet_mode or 0.8 for full_mode.
sigma_gene	(default TRUE) if TRUE, fits gene specific overdispersion parameter. If FALSE, overdispersion parameter is same across all genes.
PRECISION.THRESHOLD	(default 0.05) for checking for convergence, the maximum parameter change per algorithm step
cell_types_present	cell types (a superset of 'cell_types') to be considered as occurring often enough to consider for gene expression contamination during the step filtering out marker genes of other cell types.
normalize_expr	(default FALSE) if TRUE, constrains total gene expression to sum to 1 in each condition.
logs	(default FALSE) if TRUE, writes progress to logs/de_logs.txt
test_error	(default FALSE) if TRUE, exits after testing for error messages without running CSIDE. If set to TRUE, this can be used to quickly evaluate if CSIDE will run without error.

Details

The design matrix contains an intercept column only. Uses maximum likelihood estimation to estimate gene expression and standard errors for each gene and each cell type.

Value

an RCTD object containing the results of the CSIDE algorithm. Contains objects de_results, which contain the results of the CSIDE algorithm including 'gene_fits', which contains the results of fits on individual genes, in addition 'sig_gene_list', a list, for each cell type, of significant genes detected by CSIDE. Additionally, the object contains 'internal_vars_de' a list of variables that are used internally by CSIDE

---

run.CSIDE.nonparam *Runs CSIDE on a RCTD object to detect nonparametric smooth gene expression patterns*

Description

Identifies cell type specific smooth gene expression patterns. The design matrix contains thin plate spline basis functions spanning the space of smooth functions. Uses maximum likelihood estimation to estimate DE and standard errors for each gene and each cell type. Selects genes with significant nonzero DE.

Usage

```
run.CSIDE.nonparam(
  myRCTD,
  df = 15,
  barcodes = NULL,
  cell_types = NULL,
  cell_type_threshold = 125,
  gene_threshold = 5e-05,
  doublet_mode = T,
  weight_threshold = NULL,
  sigma_gene = T,
  PRECISION.THRESHOLD = 0.05,
  cell_types_present = NULL,
  fdr = 0.01,
  test_genes_sig = T,
  logs = F,
  test_error = F
)
```

Arguments

myRCTD	an RCTD object with annotated cell types e.g. from the run.RCTD function.
df	(default 15) the degrees of freedom, or number of basis functions to be used in the model.
barcodes	the barcodes, or pixel names, of the SpatialRNA object to be used when fitting the model.

`cell_types` the cell types used for CSIDE. If null, cell types will be chosen with aggregate occurences of at least 'cell_type_threshold', as aggregated by `aggregate_cell_types`

`cell_type_threshold` (default 125) min occurence of number of cells for each cell type to be used, as aggregated by `aggregate_cell_types`

`gene_threshold` (default 5e-5) minimum average normalized expression required for selecting genes

`doublet_mode` (default TRUE) if TRUE, uses RCTD doublet mode weights. Otherwise, uses RCTD full mode weights

`weight_threshold` (default NULL) the threshold of total normalized weights across all cell types in `cell_types` per pixel to be included in the model. Default 0.99 for doublet_mode or 0.8 for full_mode.

`sigma_gene` (default TRUE) if TRUE, fits gene specific overdispersion parameter. If FALSE, overdispersion parameter is same across all genes.

`PRECISION.THRESHOLD` (default 0.05) for checking for convergence, the maximum parameter change per algorithm step

`cell_types_present` cell types (a superset of 'cell_types') to be considered as occuring often enough to consider for gene expression contamination during the step filtering out marker genes of other cell types.

`fdr` (default 0.01) false discovery rate for hypothesis testing

`test_genes_sig` (default TRUE) logical controlling whether genes will be tested for significance

`logs` (default FALSE) if TRUE, writes progress to logs/de_logs.txt

`test_error` (default FALSE) if TRUE, exits after testing for error messages without running CSIDE. If set to TRUE, this can be used to quickly evaluate if CSIDE will run without error.

Value

an `RCTD` object containing the results of the CSIDE algorithm. Contains objects `de_results`, which contain the results of the CSIDE algorithm including 'gene_fits', which contains the results of fits on individual genes, in addition 'sig_gene_list', a list, for each cell type, of significant genes detected by CSIDE. Additionally, the object contains 'internal_vars_de' a list of variables that are used internally by CSIDE

`run.CSIDE.regions` *Runs CSIDE on a `RCTD` object for DE across multiple discrete regions*

Description

Identifies cell type specific differential expression (DE) across multiple discrete regions The design matrix contains for each region a column of 0s and 1s representing membership in that region. Uses maximum likelihood estimation to estimate DE and standard errors for each gene and each cell type. Selects genes with significant nonzero DE. Tests for differences in gene expression across regions.

Usage

```
run.CSIDE.regions(
  myRCTD,
  region_list,
  cell_types = NULL,
  cell_type_threshold = 125,
  gene_threshold = 5e-05,
  doublet_mode = T,
  weight_threshold = NULL,
  sigma_gene = T,
  PRECISION.THRESHOLD = 0.05,
  cell_types_present = NULL,
  fdr = 0.01,
  test_genes_sig = T,
  logs = F,
  log_fc_thresh = 0.4,
  test_error = F
)
```

Arguments

myRCTD	an RCTD object with annotated cell types e.g. from the run.RCTD function.
region_list	a list of character vectors, where each vector contains pixel names, or barcodes, for a single region. These pixel names should be a subset of the pixels in the SpatialRNA object
cell_types	the cell types used for CSIDE. If null, cell types will be chosen with aggregate occurences of at least 'cell_type_threshold', as aggregated by aggregate_cell_types
cell_type_threshold	(default 125) min occurence of number of cells for each cell type to be used, as aggregated by aggregate_cell_types
gene_threshold	(default 5e-5) minimum average normalized expression required for selecting genes
doublet_mode	(default TRUE) if TRUE, uses RCTD doublet mode weights. Otherwise, uses RCTD full mode weights
weight_threshold	(default NULL) the threshold of total normalized weights across all cell types in cell_types per pixel to be included in the model. Default 0.99 for doublet_mode or 0.8 for full_mode.
sigma_gene	(default TRUE) if TRUE, fits gene specific overdispersion parameter. If FALSE, overdispersion parameter is same across all genes.
PRECISION.THRESHOLD	(default 0.05) for checking for convergence, the maximum parameter change per algorithm step
cell_types_present	cell types (a superset of 'cell_types') to be considered as occuring often enough to consider for gene expression contamination during the step filtering out marker genes of other cell types.
fdr	(default 0.01) false discovery rate for hypothesis testing
test_genes_sig	(default TRUE) logical controlling whether genes will be tested for significance

logs	(default FALSE) if TRUE, writes progress to logs/de_logs.txt
log_fc_thresh	(default 0.4) the natural log fold change cutoff for differential expression
test_error	(default FALSE) if TRUE, exits after testing for error messages without running CSIDE. If set to TRUE, this can be used to quickly evaluate if CSIDE will run without error.

Value

an RCTD object containing the results of the CSIDE algorithm. Contains objects de_results, which contain the results of the CSIDE algorithm including 'gene_fits', which contains the results of fits on individual genes, in addition 'sig_gene_list', a list, for each cell type, of significant genes detected by CSIDE. Additionally, the object contains 'internal_vars_de' a list of variables that are used internally by CSIDE

---

run.CSIDE.replicates *Runs CSIDE on a RCTD.replicates object*

Description

Identifies cell type specific differential expression (DE) as a function of the explanatory variable for each replicate. The design matrix contains an intercept column and a column of the explanatory variable. Uses maximum likelihood estimation to estimate DE and standard errors for each gene and each cell type. Selects genes with significant nonzero DE. Note: a minimum of three replicates are required for population mode.

Usage

```
run.CSIDE.replicates(
  RCTD.replicates,
  cell_types,
  explanatory.variable.replicates = NULL,
  X.replicates = NULL,
  cell_type_threshold = 125,
  gene_threshold = 5e-05,
  doublet_mode = T,
  weight_threshold = NULL,
  sigma_gene = T,
  PRECISION.THRESHOLD = 0.05,
  cell_types_present = NULL,
  fdr = 0.01,
  population_de = F,
  replicate_index = NULL,
  normalize_expr = F,
  test_genes_sig_individual = F,
  de_mode = "single",
  df = 15,
  barcodes = NULL,
  log_fc_thresh = 0.4,
  test_error = F,
  params_to_test = NULL,
  test_mode = "individual"
)
```

Arguments

RCTD.replicates	an RCTD.replicates object with annotated cell types e.g. from the run.RCTD.replicates function.
cell_types	the cell types used for CSIDE. Each cell type must occur at least 'cell_type_threshold', as aggregated by aggregate_cell_types
explanatory.variable.replicates	(only used for de_mode = single) a list of the named numeric vectors representing for each replicate the explanatory variable used for explaining differential expression in CSIDE. Names of the vectors are the SpatialRNA pixel names, and values should be standardized between 0 and 1.
X.replicates	(only used for de_mode = general) a list for each replicate of matrices containing the covariates for running CSIDE. The rownames represent pixel names and should be a subset of the pixels in the SpatialRNA object. The columns each represent a covariate for explaining differential expression and need to be linearly independent.
cell_type_threshold	(default 125) min occurence of number of cells for each cell type to be used, as aggregated by aggregate_cell_types
gene_threshold	(default 5e-5) minimum average normalized expression required for selecting genes
doublet_mode	(default TRUE) if TRUE, uses RCTD doublet mode weights. Otherwise, uses RCTD full mode weights
weight_threshold	(default NULL) the threshold of total normalized weights across all cell types in cell_types per pixel to be included in the model. Default 0.99 for doublet_mode or 0.95 for full_mode.
sigma_gene	(default TRUE) if TRUE, fits gene specific overdispersion parameter. If FALSE, overdispersion parameter is same across all genes.
PRECISION.THRESHOLD	(default 0.05) for checking for convergence, the maximum parameter change per algorithm step
cell_types_present	cell types (a superset of 'cell_types') to be considered as occuring often enough to consider for gene expression contamination during the step filtering out marker genes of other cell types.
fdr	(default 0.01) false discovery rate for hypothesis testing
population_de	(default FALSE) whether population-level DE should be run (can also be run later using the CSIDE.population.inference function.)
replicate_index	(default all replicates) integer list of replicate indices (subset of 1:N_replicates) to be run for CSIDE
normalize_expr	(default FALSE) if TRUE, constrains total gene expression to sum to 1 in each condition.
test_genes_sig_individual	(default FALSE) logical controlling whether on individual samples genes will be tested for significance.
de_mode	(default 'single', otherwise 'nonparam' or 'general') if 'single', calls run.CSIDE.single. If 'nonparam', calls run.CSIDE.nonparam. If 'general', calls run.CSIDE.

df	(default 15) for de_mode = nonparam, the degrees of freedom, or number of basis functions to be used in the model.
barcodes	for de_mode = nonparam, the barcodes, or pixel names, of the SpatialRNA object to be used when fitting the model.
log_fc_thresh	(default 0.4) the natural log fold change cutoff for differential expression
test_error	(default FALSE) if TRUE, first tests for error messages before running CSIDE. If set to TRUE, this can be used to quickly evaluate if CSIDE will run without error.
params_to_test:	(default 2 for test_mode = 'individual', all parameters for test_mode = 'categorical'). An integer vector of parameter indices to test. For example c(1,4,5) would test only parameters corresponding to columns 1, 4, and 5 of the design matrix.

Value

an RCTD.replicates object containing the results of the CSIDE algorithm. See RCTD.replicates for documentation on the population_de_results, population_sig_gene_list, and population_sig_gene_df objects.

---

run.CSIDE.single *Runs CSIDE on a RCTD object with a single explanatory variable*

Description

Identifies cell type specific differential expression (DE) as a function of the explanatory variable. The design matrix contains an intercept column and a column of the explanatory variable. Uses maximum likelihood estimation to estimate DE and standard errors for each gene and each cell type. Selects genes with significant nonzero DE.

Usage

```
run.CSIDE.single(
  myRCTD,
  explanatory.variable,
  cell_types = NULL,
  cell_type_threshold = 125,
  gene_threshold = 5e-05,
  doublet_mode = T,
  weight_threshold = NULL,
  sigma_gene = T,
  PRECISION.THRESHOLD = 0.05,
  cell_types_present = NULL,
  fdr = 0.01,
  test_genes_sig = T,
  normalize_expr = F,
  logs = F,
  log_fc_thresh = 0.4,
  test_error = F,
  fdr_method = "BH"
)
```

Arguments

myRCTD	an RCTD object with annotated cell types e.g. from the run.RCTD function.
explanatory.variable	a named numeric vector representing the explanatory variable used for explaining differential expression in CSIDE. Names of the variable are the SpatialRNA pixel names, and values should be standardized between 0 and 1.
cell_types	the cell types used for CSIDE. If null, cell types will be chosen with aggregate occurrences of at least 'cell_type_threshold', as aggregated by aggregate_cell_types
cell_type_threshold	(default 125) min occurrence of number of cells for each cell type to be used, as aggregated by aggregate_cell_types
gene_threshold	(default 5e-5) minimum average normalized expression required for selecting genes
doublet_mode	(default TRUE) if TRUE, uses RCTD doublet mode weights. Otherwise, uses RCTD full mode weights
weight_threshold	(default NULL) the threshold of total normalized weights across all cell types in cell_types per pixel to be included in the model. Default 0.99 for doublet_mode or 0.8 for full_mode.
sigma_gene	(default TRUE) if TRUE, fits gene specific overdispersion parameter. If FALSE, overdispersion parameter is same across all genes.
PRECISION.THRESHOLD	(default 0.05) for checking for convergence, the maximum parameter change per algorithm step
cell_types_present	cell types (a superset of 'cell_types') to be considered as occurring often enough to consider for gene expression contamination during the step filtering out marker genes of other cell types.
fdr	(default 0.01) false discovery rate for hypothesis testing
test_genes_sig	(default TRUE) logical controlling whether genes will be tested for significance
normalize_expr	(default FALSE) if TRUE, constrains total gene expression to sum to 1 in each condition.
logs	(default FALSE) if TRUE, writes progress to logs/de_logs.txt
log_fc_thresh	(default 0.4) the natural log fold change cutoff for differential expression
test_error	(default FALSE) if TRUE, exits after testing for error messages without running CSIDE. If set to TRUE, this can be used to quickly evaluate if CSIDE will run without error.
fdr_method	(default BH) if BH, uses the Benjamini-Hochberg method. Otherwise, uses local fdr with an empirical null.

Value

an RCTD object containing the results of the CSIDE algorithm. Contains objects de_results, which contain the results of the CSIDE algorithm including 'gene_fits', which contains the results of fits on individual genes, in addition 'sig_gene_list', a list, for each cell type, of significant genes detected by CSIDE. Additionally, the object contains 'internal_vars_de' a list of variables that are used internally by CSIDE

run.RCTD *Runs the RCTD pipeline on a RCTD object*

Description

Equivalent to sequentially running the functions fitBulk, choose_sigma_c, and fitPixels

Usage

```
run.RCTD(RCTD, doublet_mode = "doublet")
```

Arguments

RCTD	an RCTD object created using the create.RCTD function.

Details

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.

Value

an RCTD object containing the results of the RCTD algorithm. Please see RCTD documentation for more information on interpreting the content of the RCTD object.

run.RCTD.replicates *Runs the RCTD pipeline on a RCTD.replicates object*

Description

For each SpatialRNA replicate in the RCTD.replicates object, runs the run.RCTD function to assign cell types.

Usage

```
run.RCTD.replicates(RCTD.replicates, doublet_mode = "doublet")
```

Arguments

RCTD.replicates	an RCTD.replicates object created using the create.RCTD.replicates function.
doublet_mode	character string, either "doublet", "multi", or "full" on which mode to run RCTD. Please see above description.

Details

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.

Value

an RCTD.replicates object containing the results of the RCTD algorithm. Please see RCTD.replicates and RCTD documentation for more information on interpreting the content of this object.

---

save.CSIDE.replicates *Saves the CSIDE population-level differential expression results for a RCTD.replicates object*

Description

First, CSIDE must have been run on all replicates at the population level using e.g. the run.CSIDE.replicates function.

Usage

```
save.CSIDE.replicates(RCTD.replicates, resultsdir)
```

Arguments

RCTD.replicates	a RCTD.replicates object containing population-level DE inference results.
resultsdir	a directory where to save the significant gene matrices for each cell type.

---

set_cell_types_assigned *If cell types have been assigned to the RCTD object, running this function will toggle the cell_types_assigned variable, which enables CSIDE to be run.*

Description

If cell types have been assigned to the RCTD object, running this function will toggle the cell_types_assigned variable, which enables CSIDE to be run.

Usage

```
set_cell_types_assigned(myRCTD)
```

Arguments

myRCTD	an RCTD object with annotated cell types from the run.RCTD function.

Value

the 'myRCTD' object with cell_types_assigned set to TRUE

set_likelihood_vars *Sets Precomputed Probabiliites as Global Variable*

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

Q_mat_loc	Matrix of precomputed probabiliites, as previously computed by get_Q_mat
X_vals	the x-values used for computing the likelihood functions.
sigma	(default NULL). If NULL, computes SQ_mat according to Q_mat_loc. Else, uses precomputed values of SQ_mat stored in SQ_mat_all with index sigma

spacexr *spacexr: an R package for assigning cell types and cell type specific differential expression to spatial transcriptomics data.*

Running RCTD

To get started, create a SpatialRNA object (called puck here) for the spatial transcriptomics data and a Reference object (called reference here) for the scRNA-seq data. Then simply run RCTD as:

```
myRCTD <- create.RCTD(puck, reference)
```

```
myRCTD <- run.RCTD(myRCTD)
```

Running CSIDE

After running RCTD, create an explanatory variable ('explanatory.variable') representing a covariate hypothesized to explain gene expression. Then, to detect cell type-specific differential expression, simply run CSIDE as:

```
myRCTD <- run.CSIDE.single(puck, explanatory.variable)
```

SpatialRNA *constructor of SpatialRNA object*

Description

constructor of SpatialRNA object

Usage

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

Arguments

coords	A data.frame (or matrix) representing the spatial pixel locations. rownames are barcodes/pixel names, and there should be two columns for 'x' and for 'y'.
counts	A matrix (or dgCmatrix) representing Digital Gene Expression (DGE). Rownames should be genes and colnames represent barcodes/pixel names.
nUMI	Optional, a named (by pixel barcode) list of total counts or UMI's appearing at each pixel. If not provided, nUMI will be assumed to be the total counts appearing on each pixel.
use_fake_coords	logical, FALSE by default. If true, the 'coords' parameter will be ignored, and replaced with a placeholder coords matrix.
require_int	logical, TRUE by default. If true, requires counts and nUMI to be integers. Counts should be untransformed count-level data

Value

Returns a SpatialRNA object containing the coordinates and counts from the input files

SpatialRNA-class *An S4 class to represent Spatial Transcriptomic data*

Description

An S4 class to represent Spatial Transcriptomic data

Slots

`coords` a dataframe with x and y coordinates of each pixel

`counts` a sparse matrix of raw counts for each gene (rowname) and each pixel (colnames or barcodes)

`n_cell_type` the number of cell types

`cell_type_names` a list of cell type names

`nUMI` a named list (by barcode) of total UMIs per pixel

`cell_labels` a factor of cell type labels for each pixel

`write_de_summary` *Saves to csv the CSIDE significant gene dataframes after running CSIDE*

Description

Saves to csv the CSIDE significant gene dataframes after running CSIDE

Usage

```
write_de_summary(myRCTD, datadir)
```

Arguments

`myRCTD`	`RCTD` object containing `de_results`, after running CSIDE
`datadir`	output directory

Index