Package: SiNMFiD (via r-universe)

August 13, 2024

Title Supervised iNMF informed Deconvolution

Version 0.0.0.9000

Description A package for completing cell type deconvolution on bulk spatial transcriptomic data utilizing multiple reference scRNA-seq datasets.

License MIT + file LICENSE

Encoding UTF-8

Language es

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Imports cowplot, ggdendro, ggplot2, grDevices, hdf5r, Matrix, matrixTests, rgl, rliger, transport, viridis, utils, lsa

Suggests cocoframer, knitr, rmarkdown

VignetteBuilder knitr

Repository https://welch-lab.r-universe.dev

RemoteUrl https://github.com/welch-lab/SiNMFiD

RemoteRef HEAD

RemoteSha 73012e2f3f981fdff6864896d9d68849f8929faf

Contents

`analyze_gene_signatures` *Calculate relationships between cell types*

Description

Calculate relationships between cell types

Usage

```
analyze_gene_signatures(
  filepath,
  analysis.name,
  spatial.data.name,
  rand.seed = 123,
  cell.types.use = NULL,
  return.objs = F
)
```

Arguments

`filepath` Path to analysis directory

`analysis.name` String identifying the analysis

`spatial.data.name`
String identifying the spatial sample

rand.seed Integer random seed

cell.types.use A string of cell type labels to include in the plot, by default all cell types present

return.objs Logical, whether to return a list of matrices of derived data

Value

named list of cosine similarity matrix and hierarchical clustering, if return.objs = TRUE

analyze_spatial_correlation

Calculate relationships between cell type distributions

Description

Calculate relationships between cell type distributions

Usage

```
analyze_spatial_correlation(
  filepath,
  analysis.name,
  spatial.data.name,
  rand.seed = 123,
  mat.use = "proportions",
  cell.types.use = NULL,
  return.objs = F
)
```

Arguments

filepath Path to analysis directory

analysis.name String identifying the analysis

spatial.data.name String identifying the spatial sample

rand.seed Integer random seed

mat.use A string, either "raw" or "proportions" referring to what version of the results to summarize

cell.types.use A string of cell type labels to include in the plot, by default all cell types present

return.objs Logical, whether to return a list of matrices of derived data

Value

named list of pearson correlation matrix and hierarchical clustering, if return.objs = TRUE

calculate_cell_sizes *Calculate cell sizes with all reference data*

Description

Calculate cell sizes with all reference data

Usage

```
calculate_cell_sizes(
  data.list,
  annotations,
  filepath,
  analysis.name,
  datasets.remove = NULL,
  plot.hist = FALSE,
  chunk = 1000
)
```

Arguments

`data.list` Various formats are allowed, including 1. a liger object; 2. a character vector containing file names to RDS/H5 files. 3. Named list of liger object, RDS/H5 file name, matrix/dgCMatrix. List option can have element types mixed. A liger object have to be of version older than 1.99. RDS files must contain base dense matrix or dgCMatrix supported by package "Matrix". H5 files must contain dataset processed by rliger < 1.99.

`annotations` Named factor of all cell type assignments, should be concatenated from all datasets.

`filepath` Path to analysis directory where output sampling needs to be stored.

`analysis.name` String identifying the analysis, used to make up a sub-folder name.

`datasets.remove` Character vector of datasets to be excluded from sampling if `data.list` is a liger object. Named list of dataset names for exluding datasets in liger objects passed with a list `data.list`. See `sample_single_cell` examples.

`plot.hist` Logical, if to display and save histograms of nUMIs by cell type

`chunk` Integer chunk size for processing sparse data stored in H5. Number of cells to load into memory per iteration. Default `1000`.

Value

Nothing is returned, but the following file will be stored to local:

- `"<filepath>/<analysis.name>/cell_size_histogram.pdf"` - A PDF file for the histogram that shows nUMI per cell distribution for each cell type
- `"<filepath>/<analysis.name>/cell_size.RDS"` - RDS file of a named numeric vector object, total number of counts per cell type across all datasets.

calculate_wasserstein *Calculate the Wasserstein distance between cell-types and genes*

Description

Calculate the Wasserstein distance between cell-types and genes

Usage

```
calculate_wasserstein(
  filepath,
  analysis.name,
  spatial.data.name,
  rand.seed = 123,
  mat.use = "proportions",
  cell.types.use = NULL,
  genes.use = NULL,
  p = 2,
  min.samples = 1,
  return.objs = F
)
```

Arguments

`filepath`	Path to analysis directory
`analysis.name`	String identifying the analysis
`spatial.data.name`	String identifying the spatial sample
`rand.seed`	Integer random seed
`mat.use`	A string, either "raw" or "proportions" referring to what version of the results to summarize
`cell.types.use`	A string of cell type labels to include in the plot, by default all cell types present
`genes.use`	A string of genes to include in a plot, by default none
`p`	The `p` exponent used for the Minkowski distance
`min.samples`	Integer value, the minimum number of samples a cell type can load on and be included in the analysis
`return.objs`	Logical, whether to return a list of matrices of derived data

Value

matrix of pairwise Wasserstein distances if `return.objs` = `TRUE`

cell_type_loading_histogram

Generate histograms of loading by cell type

Description

Generate histograms of loading by cell type

Usage

```
cell_type_loading_histogram(
  filepath,
  analysis.name,
  spatial.data.name,
  rand.seed = 123,
  mat.use = "proportions",
  cell.types.plot = NULL,
  print.plots = TRUE,
  bin.num = 30
)
```

Arguments

`filepath`	Path to analysis directory
`analysis.name`	String identifying the analysis
`spatial.data.name`	String identifying the spatial sample
`rand.seed`	Integer random seed
`mat.use`	A string, either "raw" or "proportions" referring to what version of the results to summarize
`print.plots`	Logical, whether to display results in the plots panel
`bin.num`	Integer number of bins to use in histogram
`cell.types.use`	A string of cell type labels to include in the plot, by default all cell types present

Value

nothing

deconvolve_spatial *Title*

Description

Title

Usage

```
deconvolve_spatial(
  filepath,
  analysis.name,
  spatial.data.name,
  rand.seed = 123,
  cell.size = T
)
```

Arguments

`filepath`	Path to analysis directory
`analysis.name`	String identifying the analysis
`spatial.data.name`	String identifying the spatial sample
`rand.seed`	Integer random seed
`cell.size`	Logical, if to scale gene signatures by cell sizes

Value

nothing

generate_label_gifs *Title*

Description

Title

Usage

```
generate_label_gifs(
  filepath,
  analysis.name,
  spatial.data.name,
  labels.plot,
  dims = c(500, 500)
)
```

Arguments

`filepath`	Path to analysis directory
`analysis.name`	String identifying the analysis
`spatial.data.name`	String identifying the spatial sample
`labels.plot`	A named vector or matrix of labels to plot for the provided coordinates
`dims`	Integer vector of length 2 corresponding to the width and height of the RGL window

Value

nothing

`generate_loading_gifs` *Generate gifs of cell type distributions derived from deconvolution in space*

Description

Generate gifs of cell type distributions derived from deconvolution in space

Usage

```
generate_loading_gifs(
  filepath,
  analysis.name,
  spatial.data.name,
  rand.seed = 123,
  mat.use = "proportions",
  cell.types.plot = NULL,
  filter = NULL,
  dims = c(500, 500)
)
```

Arguments

`filepath`	Path to analysis directory
`analysis.name`	String identifying the analysis
`spatial.data.name`	String identifying the spatial sample
`rand.seed`	Integer random seed
`mat.use`	A string, either "raw" or "proportions" referring to what version of the results to summarize
`cell.types.plot`	A character vector of cell types to plot
`dims`	Integer vector of length 2 corresponding to the width and height of the RGL window

Value

nothing

learn_gene_signatures *Title*

Description

Title

Usage

```
learn_gene_signatures(
  filepath,
  analysis.name,
  spatial.data.name,
  rand.seed = 123,
  lambda = 1,
  thresh = 1e-08,
  max.iters = 100,
  nrep = 1,
  print.obj = FALSE,
  verbose = FALSE
)
```

Arguments

`filepath` Path to analysis directory

`analysis.name` String identifying the analysis

`spatial.data.name` String identifying the spatial sample

`rand.seed` Integer random seed

`lambda` Double, regularization parameter for which increasing penalizes dataset-specific effects

`thresh` Double, minimum fractional change in objective function to continue iteration

`max.iters` Integer maximum of iterations to complete before pausing

`nrep` Number of random starts to complete

`print.obj` Logical, if to print current value of objective

`verbose` Logical, if to print the final objective and best random seed

Value

nothing

`load_objs` *Load data from one of multiple formats*

Description

Load data from one of multiple formats

Usage

```
load_objs(objs, datasets.remove)
```

Arguments

`objs` A named list of matrices (dgCMatrix), RDS file paths to matirces, H5 file paths to LIGER analyzed datasets.

Value

list object. List element type depends on input.

`mirror_spatial_coords` *Flip axes in spatial data*

Description

Flip axes in spatial data

Usage

```
mirror_spatial_coords(
  filepath,
  analysis.name,
  spatial.data.name,
  axes.flip = c(FALSE, FALSE, FALSE),
  overwrite = T
)
```

Arguments

`filepath` Path to analysis directory

`analysis.name` String identifying the analysis

`spatial.data.name`
String identifying the spatial sample

`axes.flip` A vector with three logicals, corresponding to which of the axes to invert

`overwrite` Logical, if the original data should be overwritten, otherwise "`spatial.data.name_mirror_x/_y,_z`is created

Value

nothing

---

overlay_subregion_gifs *Title*

Description

Title

Usage

```
overlay_subregion_gifs(
  filepath,
  analysis.name,
  spatial.data.name,
  rand.seed = 123,
  mat.use = "proportions",
  cell.types.plot = NULL,
  subregions.plot = NULL,
  filter = NULL,
  dims = c(500, 500)
)
```

Arguments

`filepath`	filepath
`analysis.name`	analysis.name
`spatial.data.name`	spatial.data.name
`rand.seed`	rand.seed
`mat.use`	mat.use
`cell.types.plot`	cell.types.plot
`subregions.plot`	subregions.plot
`filter`	filter
`dims`	dims

Value

nothing

plot_analyze_gene_signatures

Plot results of `analyze_gene_signatures`

Description

Plot results of `analyze_gene_signatures`

Usage

```
plot_analyze_gene_signatures(
  filepath,
  analysis.name,
  spatial.data.name,
  rand.seed = 123,
  print.plots = T
)
```

Arguments

`filepath`	Path to analysis directory
`analysis.name`	String identifying the analysis
`spatial.data.name`	String identifying the spatial sample
`rand.seed`	Integer random seed
`print.plots`	Logical, whether to display results in the plots panel

Value

nothing

plot_analyze_spatial_correlation

Plot results of `analyze_spatial_correlation`

Description

Plot results of `analyze_spatial_correlation`

Usage

```
plot_analyze_spatial_correlation(
  filepath,
  analysis.name,
  spatial.data.name,
  rand.seed = 123,
  print.plots = TRUE
)
```

Arguments

`filepath` Path to analysis directory

`analysis.name` String identifying the analysis

`spatial.data.name`
String identifying the spatial sample

`rand.seed` Integer random seed

`print.plots` Logical, whether to display results in the plots panel

Value

nothing

`plot_calculate_wasserstein` *Plot results of* `calculate_wasserstein`

Description

Plot results of `calculate_wasserstein`

Usage

```
plot_calculate_wasserstein(
  filepath,
  analysis.name,
  spatial.data.name,
  rand.seed = 123,
  print.plots = T
)
```

Arguments

`filepath`	Path to analysis directory
`analysis.name`	String identifying the analysis
`spatial.data.name`	String identifying the spatial sample
`rand.seed`	Integer random seed
`print.plots`	Logical, whether to display results in the plots panel

Value

nothing

`plot_summarize_by_layer` *Plot results of* `summarize_by_layer`

Description

Plot results of `summarize_by_layer`

Usage

```
plot_summarize_by_layer(
  filepath,
  analysis.name,
  spatial.data.name,
  rand.seed = 123,
  print.plots = T
)
```

Arguments

`filepath`	Path to analysis directory
`analysis.name`	String identifying the analysis
`spatial.data.name`	String identifying the spatial sample
`rand.seed`	Integer random seed
`print.plots`	Logical, whether to display results in the plots panel

Value

nothing

qc_spatial_data *Quality-control spatial data*

Description

Quality-control spatial data

Usage

```
qc_spatial_data(
  filepath,
  analysis.name,
  spatial.data.name,
  count.data = FALSE,
  z = 1,
  n.umi.thresh = 150,
  rand.seed = 123
)
```

Arguments

`filepath`	Path to analysis directory
`analysis.name`	String identifying the analysis
`spatial.data.name`	String identifying the spatial sample
`count.data`	Logical, if the spatial data is from a counts or intensity-based modality
`z`	Double, the standard deviations above the mean that the number of NAs in a gene can be before the gene is removed, for intensity data
`n.umi.thresh`	Integer number of counts below which to remove a sample, for counts based data
`rand.seed`	Integer random seed

Value

nothing

reference_3d_coordinates

Generate silouhettes of the data along all three axes

Description

Generate silouhettes of the data along all three axes

Usage

```
reference_3d_coordinates(
  filepath,
  analysis.name,
  spatial.data.name,
  save.plots = FALSE
)
```

Arguments

filepath	Path to analysis directory
analysis.name	String identifying the analysis
spatial.data.name	String identifying the spatial sample
save.plots	A logical, corresponding with if to save requested plots upon generation

Value

nothing

register_voxel_to_label

Transfer labels from coarse-grained sampled

Description

Transfer labels from coarse-grained sampled

Usage

```
register_voxel_to_label(
  filepath,
  analysis.name,
  spatial.data.name,
  labels.use,
  label.name
)
```

Arguments

`filepath` Path to analysis directory

`analysis.name` String identifying the analysis

`spatial.data.name` String identifying the spatial sample

`labels.use` Named vector of labels for the prevoxelized data

`label.name` String identifying the label set

Value

nothing

`sample_single_cell` *Sample from single cell reference datasets*

Description

Sample from single cell reference datasets

Usage

```
sample_single_cell(
  data.list,
  annotations,
  filepath,
  analysis.name,
  datasets.remove = NULL,
  n.cells = 500,
  rand.seed = 123,
  chunk = 1000
)
```

Arguments

`data.list` Various formats are allowed, including 1. a liger object; 2. a character vector containing file names to RDS/H5 files. 3. Named list of liger object, RDS/H5 file name, matrix/dgCMatrix. List option can have element types mixed. A liger object have to be of version older than 1.99. RDS files must contain base dense matrix or dgCMatrix supported by package "Matrix". H5 files must contain dataset processed by rliger < 1.99.

`annotations` Named factor of cell type assignments.

`filepath` Path to analysis directory where output sampling needs to be stored.

`analysis.name` String identifying the analysis, used to make up a sub-folder name.

`datasets.remove`
Character vector of datasets to be excluded from sampling if `data.list` is a liger object. Named list of dataset names for exluding datasets in liger objects passed with a list `data.list`.

`n.cells` Integer value corresponding to maximum number of samples per cell type. Default `500`.

`rand.seed` Integer random seed for reproducible sampling.

`chunk` Integer chunk size for processing sparse data stored in H5. Number of cells to load into memory per iteration. Default `1000`.

Value

Nothing is returned. File `"norm_data.RDS"` will be stored under `"<filepath>/<analysis.name>/<rand.seed>/"`, containing a list of downsampled scaled (not centered) data matrix. File `"sampled_cells.RDS"` is stored at the same path, containing barcode vector of the sampled cells. File `"source_annotations.RDS"` is stored at `"<filepath>/<analysis.name>/"` which contains input `annotations`.

Examples

```
## Not run:
# Explanation for how `datasets.remove` works with example:

names(lig@raw.data)
# above should show "data1", "data2", "data3", ...
# Then when sampling from `lig`, the first two datasets can be excluded with
sample_single_cell(data.list = lig, datasets.remove = c("data1", "data2"))

# If we got a list of liger object
sample_single_cell(data.list = list(human = lig1, mouse = lig2),
                   datasets.remove = list(human = c("data1", "data2"),
                                          mouse = c("10x1")))

## End(Not run)
```

`save_spatial_data` *Add a new spatial dataset to the analysis directory*

Description

Add a new spatial dataset to the analysis directory

Usage

```
save_spatial_data(
  filepath,
  analysis.name,
  spatial.data.file,
  coords.file,
  spatial.data.name
)
```

Arguments

filepath Path to analysis directory

analysis.name String identifying the analysis

spatial.data.file
Path to an RDS file containing desired expression data

coords.file Path to an RDS file containing desired coordinate data

spatial.data.name
String identifying the spatial sample

Value

nothing

---

select_defining_genes *select variable genes with the Kruskal-Wallis test*

Description

select variable genes with the Kruskal-Wallis test

Usage

```
select_defining_genes(
  filepath,
  analysis.name,
  deconv.gene.num = 2000,
  gene.num.tol = 50,
  rand.seed = 123
)
```

Arguments

filepath Path to analysis directory

analysis.name String identifying the analysis

deconv.gene.num
Integer, the number of genes to select

gene.num.tol Integer, the maximum difference between the number of genes selected and deconv.gene.num

rand.seed Integer random seed

Value

nothing

start_analysis *Set up new analysis directory*

Description

Set up new analysis directory

Usage

```
start_analysis(filepath, analysis.name)
```

Arguments

`filepath`	Path to analysis directory
`analysis.name`	String identifying the analysis

Value

nothing

subset_spatial_data *Subset a spatial dataset by coordinates for analysis*

Description

Subset a spatial dataset by coordinates for analysis

Usage

```
subset_spatial_data(
  filepath,
  analysis.name,
  spatial.data.name,
  subset.specs = list(c(NaN, NaN), c(NaN, NaN), c(NaN, NaN)),
  new.spatial.data.name = NULL,
  out.filepath = NULL
)
```

Arguments

`filepath`	Path to analysis directory
`analysis.name`	String identifying the analysis
`spatial.data.name`	String identifying the spatial sample

`subset.specs` A list with length equal to the number of axes, with each entry a vector of length two, with the first element being the minimum value to include and the second being the maximum, or NaN to indicate a missing value

`new.spatial.data.name` String, optional name for new analysis, otherwise the default "`spatial.data.name`*subset*`n.samples`" is used

`out.filepath` Path to directory to save subset data to if not within the analysis

Value

nothing

`summarize_by_layer` *Summarize cell-type and gene expression data by*

Description

Summarize cell-type and gene expression data by

Usage

```
summarize_by_layer(
  filepath,
  analysis.name,
  spatial.data.name,
  rand.seed = 123,
  layer.list,
  type = "mean",
  mat.use = "proportions",
  cell.types.use = NULL,
  genes.use = NULL,
  return.objs = FALSE
)
```

Arguments

`filepath` Path to analysis directory

`analysis.name` String identifying the analysis

`spatial.data.name` String identifying the spatial sample

`rand.seed` Integer random seed

`layer.list` A named list of spatial samples by layer of interest

`type` A string, either "mean" or "sum", how results should be combined for summary

`mat.use` A string, either "raw", "proportions", or "assignments" referring to what version of the results to summarize

`cell.types.use`	A string of cell type labels to include in the plot, by default all cell types present
`genes.use`	A string of genes to include in a plot, by default none
`return.objs`	Logical, whether to return a list of matrices of derived data

Value

cell-type and gene expression data summarized by layer in a named list, if `return.objs = TRUE`

`summarize_clusters` *Summarize cell types present in the source annotations*

Description

Summarize cell types present in the source annotations

Usage

```
summarize_clusters(filepath, analysis.name, return.objs = F)
```

Arguments

`filepath`	Path to analysis directory
`analysis.name`	String identifying the analysis
`return.objs`	Logical, whether to return a vector of the names of clusters

Value

A vector of unique clusters in the source annotations, if `return.objs = TRUE`

`summarize_subregions` *Summarize subregions of a vector of regions of interest*

Description

Summarize subregions of a vector of regions of interest

Usage

```
summarize_subregions(
  regions,
  ontology.file = "Downloads/allen_structure_ontology.csv",
  return.objs = F
)
```

Arguments

`regions` A vector of region names

`ontology.file` A csv describing the Allen structure ontology

`return.objs` Logical, whether to return acronyms for all subregions found

Value

A vector of unique subregions within the provided regions, if `return.objs = TRUE`

`transform_coords_to_ccf`
Use predefined transformations to match some modalities to the Allen CCF

Description

Use predefined transformations to match some modalities to the Allen CCF

Usage

```
transform_coords_to_ccf(filepath, analysis.name, spatial.data.name, ish = T)
```

Arguments

`filepath` Path to analysis directory

`analysis.name` String identifying the analysis

`spatial.data.name`
String identifying the spatial sample

`ish` Logical, if the data comes from the Allen Institute quantified ISH dataset

Value

nothing

view_in_rgl *Title*

Description

Title

Usage

```
view_in_rgl(
  filepath,
  analysis.name,
  spatial.data.name,
  rand.seed = 123,
  cell.type,
  mat.use = "proportions",
  filter.samples = NULL,
  dims = c(500, 500)
)
```

Arguments

filepath	Path to analysis directory
analysis.name	String identifying the analysis
spatial.data.name	String identifying the spatial sample
rand.seed	Integer random seed
cell.type	A string corresponding to one cell type found in the deconvolution results
mat.use	A string, either "raw" or "proportions" referring to what version of the results to summarize
filter.samples	Value for binarizing results, either presence above the provided threshold or absence below
dims	Integer vector of length 2 corresponding to the width and height of the RGL window

Value

nothing

`voxelize_single_cells` *Coarse-grain spatial data to a predetermined resolution*

Description

Coarse-grain spatial data to a predetermined resolution

Usage

```
voxelize_single_cells(
  filepath,
  analysis.name,
  spatial.data.name,
  voxel.size,
  out.filepath = NULL,
  verbose = TRUE
)
```

Arguments

`filepath`	Path to analysis directory
`analysis.name`	String identifying the analysis
`spatial.data.name`	String identifying the spatial sample
`voxel.size`	Integer, side length of one voxel
`out.filepath`	Path to directory to save subset data to if not within the analysis
`verbose`	Logical, if to print several lines of metadata on results

Value

nothing

Index