Package 'spammR'

October 30, 2026

Title SPatial Analysis of Multiomics Measurements in R

Version 0.99.26

Description SpammR provides a suite of tools and a data structure to analyze diverse multiomics measurements in a spatial context. It enables the analysis and visualization of omics measurements paired with a specific image, each with spatial coordinates.

SystemRequirements python = 3.11

Config/reticulate list(packages = list(list(package = ``numpy''), list(package = ``pandas''), list(package = ``metaspace2020'') ), python_version = ``3.11'')

License MIT + file LICENSE

BugReports https://github.com/PNNL-CompBio/spammR/issues

URL https://pnnl-compbio.github.io/spammR/

Encoding UTF-8

Roxygen list(markdown = TRUE)

Suggests testthat (>= 3.0.0), BiocStyle, knitr

Config/testthat/edition 3

VignetteBuilder knitr

biocViews Spatial, Pathways, Proteomics, DifferentialExpression, Metabolomics, SystemsBiology, GeneExpression, GeneSetEnrichment, Software,Transcription,Sequencing, Microarray, Epigenetics, Cheminformatics, Transcriptomics,Coverage, Normalization, Preprocessing, Bayesian, Clustering, Regression, ExonArray, RNASeq, OneChannel, TwoChannel, MicroRNAArray, mRNAMicroarray,ProprietaryPlatforms, SingleCell, CellBiology, BiomedicalInformatics,FunctionalGenomics, ImmunoOncology, AlternativeSplicing, DifferentialSplicing,BatchEffect, MultipleComparison, QualityControl, TimeCourse, DataImport

Depends SummarizedExperiment, SingleCellExperiment, SpatialExperiment, limma, R (>= 4.5.0)

Imports dplyr, ggplot2, plotly, ggnewscale, ggpubr, png, reticulate, impute, matrixStats, scales, spdep, ggrepel, IRanges, tibble, tidyr, leapR, tidygraph, ggraph,BiocFileCache, terra, readxl, tidygraph, tidyverse, magrittr, sf, cowplot, DT

NeedsCompilation no

Config/roxygen2/version 8.1.0

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

git_branch devel

git_last_commit 6a00912

git_last_commit_date 2026-09-04

Repository Bioconductor 3.24

Date/Publication 2026-10-29

Author Sara Gosline [cre, aut] (ORCID:
<https://orcid.org/0000-0002-6534-4774>),
Harkirat Sohi [aut] (ORCID: <https://orcid.org/0009-0000-9935-1641>),
Yannick Mahlich [aut] (ORCID: <https://orcid.org/0000-0002-0717-2927>)

Maintainer Sara Gosline <sara.gosline@pnnl.gov>

Contents

spammR-package . . . 2
calc_centers . . . 3
calc_diff_pairwiseSamples . . . 4
calc_missingness . . . 4
calc_spatial_diff_ex . . . 5
convert_to_spe . . . 6
distance_based_analysis . . . 8
enrich_gradient . . . 9
enrich_ora . . . 11
impute_spe . . . 12
oneKSoilsCoords . . . 14
oneKSoilsMeta . . . 14
pancMeta . . . 14
protMeta . . . 15
retrieve_metaspace_data . . . 15
smallPancData . . . 16
spatial_heatmap . . . 17
spatial_network . . . 19
spat_reduce . . . 20
split_spe . . . 21
volcano_plot . . . 22

Index **24**

`spammR-package` *spammR: SPatial Analysis of Multiomics Measurements in R*

Description

SpammR provides a suite of tools and a data structure to analyze diverse multiomics measurements in a spatial context. It enables the analysis and visualization of omics measurements paired with a specific image, each with spatial coordinates.

Author(s)

Maintainer: Sara Gosline <sara.gosline@pnnl.gov> (ORCID)

Authors:

- Sara Gosline <sara.gosline@pnnl.gov> (ORCID)
- Harkirat Sohi <harkirat.sohi@pnnl.gov> (ORCID)
- Yannick Mahlich <yannick.mahlich@pnnl.gov> (ORCID)

See Also

Useful links:

- https://pnnl-compbio.github.io/spammR/
- Report bugs at https://github.com/PNNL-CompBio/spammR/issues

---

calc_centers *Calculate centers of spots/samples for distance based analysis*

Description

calc_centers calculates the centers of spots for numerical analysis.Currently not used.

Usage

```
calc_centers(spe)
```

Arguments

spe	SpatialExperiment object containing omics data

Value

Data frame of new coordinates

Examples

```
data(pancMeta)
data(protMeta)
data(smallPancData)
img0.spe <- convert_to_spe(smallPancData$Image_0,
  pancMeta,
  protMeta,
  feature_meta_colname = "pancProts",
  image_files = system.file("extdata", "Image_0.png", package = "spammR"),
  sample_id = "Image0",
  spatial_coords_colnames = c("x_pixels", "y_pixels"),
  image_sample_ids = "Image0", image_ids = "Image0"
)
## can't actually call internal functuon
```

`calc_diff_pairwiseSamples` *Helper function for computing pairwise differences between sample values*

Description

This function is called in distance_based_analysis()

Usage

```
calc_diff_pairwiseSamples(feature_values)
```

Arguments

`feature_values`	A 1 x n vector containing values for a feature for n

Value

Data frame of differences samples. names(feature_values) should contain names of samples

`calc_missingness` *Calculate degree of missingness for each featurea and sample*

Description

calc_missingness() calculates the fraction of samples that are missing data for each feature, and fraction of each feature that are missing for each sample.

Usage

```
calc_missingness(spe, group_name = NULL)
```

Arguments

`spe`	SpatialExperiment object
`group_name`	Name of group to use as denominator

Value

SpatialExperiment object with missingFeature column added to colData and missingSample added to rowData

Examples

```
library(spammR)
data(smallPancData)
data(pancMeta)
data(protMeta)
pooledPanc <- dplyr::bind_cols(smallPancData)
panc.spe <- convert_to_spe(pooledPanc, pancMeta, protMeta,
          feature_meta_colname = "pancProts")
calc_missingness(panc.spe)
```

`calc_spatial_diff_ex` *Identify differentially abundant features across image*

Description

`calc_spatial_diff_ex()` Calculates differential expression analysis using annotations in a SpatialExperiment object. Differential expression is calculated using functionality from the `limma` package.

Usage

```
calc_spatial_diff_ex(
  spe,
  assay_name = "proteomics",
  count_based = FALSE,
  log_transformed = FALSE,
  category_col,
  compare_vals
)
```

Arguments

`spe`	Spatial Experiment object containing data to be used for differential expression analysis
`assay_name`	Name of the dataset stored in the spe object, that is to be used for the differential expression analysis. Example: znormalized_log2
`count_based`	Set to TRUE of the data are count based, e.g. RNA-Seq
`log_transformed`	Is the data given in spe log2 transformed TRUE or FALSE
`category_col`	Name of the column that specifies category of each sample. Example: "IsletOrNot" #Categories from `category_col` will be compared in the differential expression analysis
`compare_vals`	A vector containing names of categories from category_col to be compared. Only required if there are more than two values in `category_col`

Value

A Spatial Experiment object containing differential expression results, stored in `rowData(diffEx.spe)` and `assays(diffEx.spe)` which contains the dataset on which differential expresssion analysis was carried out

Examples

```
data(smallPancData)
data(pancMeta)
data(protMeta)
pooledData <- dplyr::bind_cols(smallPancData)
pooled.panc.spe <- convert_to_spe(pooledData,
  pancMeta,
  protMeta,
  feature_meta_colname = "pancProts"
)
diffex.spe <- calc_spatial_diff_ex(pooled.panc.spe,
  category_col = "IsletOrNot"
)
```

---

convert_to_spe *Create a* SpatialExperiment *object from data*

Description

convert_to_spe() Puts omics data (omics measurements, metadata and image(s) corresponding to samples' tissue) into a SpatialExperiment (SPE) object. Most spammR functions require the input data to be an SPE object.

Usage

```
convert_to_spe(
  dat,
  sample_meta,
  feature_meta,
  assay_name = "proteomics",
  sample_id = "sample",
  sample_colname = NULL,
  feature_meta_colname = NULL,
  spatial_coords_colnames = NULL,
  rescale_image = FALSE,
  image_files = NULL,
  image_ids = NULL,
  image_sample_ids = rep(sample_id, length(image_files))
)
```

Arguments

dat	Matrix or data frame with omics measurements. Rows are features, columns are samples
sample_meta	Data frame of metadata with either sample_colname or rownames as samples (if empty)
feature_meta	Data frame of metadata with either sample_colname or rownames as samples (if empty)
assay_name	Name to be given to the data in omics_measurements_file. Example: "abundance", "log2", "znormalized_log2" or any other descriptive name

`sample_id` Name of sample, defaults to "sample01"

`sample_colname` Column name in `sample_meta` table whose entries contain sample identifiers provided as column names in `dat`.

`feature_meta_colname`
Name of column in `feature_meta`, that is to be used for identifying features. If missing defaults to rownames, which should match rownames of `dat`

`spatial_coords_colnames`
A list containing names of columns in `meta_dat` that are spatial coordinates. Default value is NULL in which case no spatial coordinates are entered

`rescale_image` A boolean set to true if you need to rescale image. Recommended when coordinates are pulled from metaspace and are known to fully cover the image

`image_files` A list containing paths of image files to be stored in the SpatialExperiment object. More images can be added later, without using this function.

`image_ids` A list containing image names/identifiers for image paths provided in image_files

`image_sample_ids`
A list of sample identifiers for each of the images provided in image_files, defaults to the value from sample_id

Value

spe.out a SpatialExperiment (SPE) object that contains all data and image(s). Ready to be used as input in spammR functions that require SPE object as input.

Examples

```
data(pancMeta)
data(protMeta)
data(smallPancData)
# We can put all samples into the same object (for statistical power)
pooledData <- dplyr::bind_cols(smallPancData)
pooled.panc.spe <- convert_to_spe(pooledData,
  pancMeta,
  protMeta,
  feature_meta_colname = "pancProts"
)

# or we can add the inmage to a single data capture
img0.spe <- convert_to_spe(smallPancData$Image_0,
  pancMeta,
  protMeta,
  sample_id = "Image0",
  feature_meta_colname = "pancProts",
  image_files = system.file("extdata", "Image_0.png", package = "spammR"),
  spatial_coords_colnames = c("x_pixels", "y_pixels"),
  image_ids = "Image0"
)
```

distance_based_analysis *Identify multiomic features correlated with an distance to an image feature*

Description

distance_based_analysis: Identifies proteins/features that show a strong correlation between spatial distance (centroid based) from a specified ROI's samples and protein/feature abundance differences between samples.

Usage

```
distance_based_analysis(
  spe,
  assay_name,
  spotHeightCol = "spot_height",
  spotWidthCol = "spot_width",
  sampleCategoryCol,
  sampleCategoryValue,
  featuresNameCol,
  corr_type = "spearman",
  corr_thresh = 0.5,
  min_samples = 5,
  allowOverlaps = TRUE
)
```

Arguments

spe	SpatialExperiment object containing spatial omics data
assay_name	Name of the assay stored in spe that is to be used for distance based analysis. Example: "znormalized_log2"
spotHeightCol	Column containing height of spot
spotWidthCol	Column containing width of spot
sampleCategoryCol	Column name in metadata (colData(spe)) that should be used for selecting samples of certain type, to define the "origin" region for distance based analysis
sampleCategoryValue	Sample category to be used for defining the "origin" region for distance based analysis
featuresNameCol	Name of column containing features (example: proteins) in rowData(spe). It is assumed that the data provided in assay(spe) is in the same order as the order in which the features are listed under the featuresNameCol
corr_type	Choose from "pearson" (default), "spearman." Correlation method to be used for calculating correlation between distance between samples and protein abundance differences. Both types of correlation provide a measure of monotonic association between two variables. Pearson is better suited for linear relationships while Spearman is better for non-linear monotonic relationships.

corr_thresh	Minimum correlation value to be used for identifying proteins that have a correlation between protein abundance differences and distance between samples. Values greater than or equal to this theshold will be used.
min_samples	of a minimum number of sample points for calculating correlation. For proteins with less than this number of sample points, correlation value is reported as NA.
allowOverlaps	allow overlaps of regions

Details

Identifies proteins/features that show a strong correlation between spatial distance and protein/feature abundance differences between a specified ROI all other ROIs (Regions Of Interest) in the dataset. Spatial distance between ROIs is calculated based on their euclidean centroid distance. Features (e.g. proteins) of interest are then extracted by correlating (Pearson or Spearman correlation) feature intensities across samples/ROIs with the previously calculated pairwise distances. NOTE: No differential expression is calculated here. For spatial differential expression please use `calc_spatial_diff_ex` & `enrich_ORA`.

Value

a `SpatialExperiment` object with the distance based data in the colData

Examples

```
data(pancMeta)
data(protMeta)
data(smallPancData)
img0.spe <- convert_to_spe(smallPancData$Image_0,
  pancMeta,
  protMeta,
  sample_id = "Image0",
  spatial_coords_colnames = c("x_pixels", "y_pixels"),
  feature_meta_colname = "pancProts",
  image_files = system.file("extdata", "Image_0.png", package = "spammR"),
  image_ids = "Image0"
)

img0.spe <- distance_based_analysis(img0.spe,
  "proteomics",
  sampleCategoryCol = "IsletOrNot",
  sampleCategoryValue = "Islet"
)
```

---

enrich_gradient *Calculate functional or pathway enrichment from a gradient*

Description

`enrich_gradient()` calculates over-representation statistics (ORA) using a ranking of genes from `distance_based_analysis()` For ORA using an external or already defined interest list of genes and gene sets, use leapR functions directly

Usage

```
enrich_gradient(
  spe,
  assay_name,
  geneset,
  feature_column,
  ranking_column,
  method = "ztest"
)
```

Arguments

spe	SpatialExperiment object containing spatial omics data and spatial diffex results
assay_name	name of assay
geneset	in GMT format
feature_column	Column of rowData that maps to gene set
ranking_column	Column of rowData that maps to ranks
method	Rank enrichmemnt method used by leapR. Default is ztest

Value

A dataframe containing results from over-representation analysis of members of gene sets in the interest list of genes based on filtering criteria above.

Examples

```
data(smallPancData)
data(pancMeta)
data(protMeta)
img0.spe <- convert_to_spe(smallPancData$Image_0,
  pancMeta,
  protMeta,
  feature_meta_colname = "pancProts",
  image_files = system.file("extdata", "Image_0.png", package = "spammR"),
  image_sample_ids = "Image0",
  spatial_coords_colnames = c("x_pixels", "y_pixels"),
  sample_id = "Image0",
  image_ids = "Image0"
)
img0.spe <- distance_based_analysis(img0.spe,
  "proteomics",
  sampleCategoryCol = "IsletOrNot",
  sampleCategoryValue = "Islet"
)
library(leapR)
data("krbpaths")
rank.res <- enrich_gradient(img0.spe,
  geneset = krbpaths,
  feature_column = "PrimaryGeneName",
  ranking_column = "IsletDistancespearmanCor"
)
head(rank.res)
```

enrich_ora *Calculate functional or pathway enrichment*

Description

enrich_ora() calculates over-representation statistics (ORA) using an interest list of genes from differential expression results in spammR and gene sets (either the ones provided in spammR or user supplied). This function uses results from calc_spatial_diff_ex. Interest list of genes for ORA is obtained from spatialDiffEx results based on the criteria specified in this function. It is a wrapper for the leapR package which is required. For ORA using an external or already defined interest list of genes and gene sets, use leapR functions directly

Usage

```
enrich_ora(
  spe,
  geneset,
  feature_column,
  pval_type_forThresh = "adjusted_pval",
  pval_thresh = 0.05,
  logFC_lowerThresh = NA,
  logFC_upperThresh = NA,
  geneset_name = "msigdb",
  sortResultsBy,
  comparison_name = ""
)
```

Arguments

spe	SpatialExperiment object containing spatial omics data and spatial diffex results
geneset	in GMT format
feature_column	Column of rowData that maps to gene set
pval_type_forThresh	Choose from "adjusted_pval" or "pval". Type of p-value that should be used for filtering statistically significant results. Default is adjusted p-value for multiple hypotheses correction.
pval_thresh	value to use for filtering based on pval_type_forThreshold. Default is 0.05. Values less than pval_thresh will be kept.
logFC_lowerThresh	Lower threshold for log Fold Change, to be used for filtering spatialDiffEx results. Default is NA
logFC_upperThresh	Upper threshold for log Fold Change, to be used for filtering spatialDiffEx results. Default is NA
geneset_name	Name of geneset provided
sortResultsBy	For sorting ORA results, choose from the following column names: "BH_pvalue" (default)
comparison_name	Example: "RSPv_vs_others" Text to indicate in results data frame, which spatial groups were compared for the interest list of genes

Value

A dataframe containing results from over-representation analysis of members of gene sets in the interest list of genes based on filtering criteria above.

Examples

```
data(smallPancData)
data(pancMeta)
data(protMeta)
pooledPanc <- dplyr::bind_cols(smallPancData)
panc.spe <- convert_to_spe(pooledPanc, pancMeta, protMeta,
feature_meta_colname = "pancProts")
diffex.spe <- calc_spatial_diff_ex(panc.spe, category_col = "IsletOrNot")
library(leapR)
data("krbpaths")
ora.res <- enrich_ora(diffex.spe, geneset = krbpaths,
         geneset_name = "krbpaths", feature_column = "PrimaryGeneName")
```

---

impute_spe *Impute missing values based on spatial coordinates*

Description

`impute_spe()` carries out imputation for missing data in the primary assay using a specified method from a range of methods. Returns additional assay in the same SPE

Usage

```
impute_spe(
  spe,
  assay_name = NULL,
  method = NULL,
  group_colname,
  k = NULL,
  protein_missingness = NULL
)
```

Arguments

`spe`	SPE containing data to be imputed.
`assay_name`	name of assay with data to be imputed
`method`	Method of imputation to be used. See details.
`group_colname`	Column name in metadata that specifies the group information to use for group_mean or knn_group. Example: ROI_abbreviation.
`k`	K value to be used for k-nearest neighbor imputation
`protein_missingness`	Proportion of samples allowed to have missing data for a protein in the given imputation method. Example: When method="global_mean," an protein_missingness of 0.5 indicates that any protein missing data for more than 50% of samples

across the entire spatial tissue covered by all samples will be excluded from the imputation method algorithm, and that protein's missing values will not be imputed. When method="group_mean," then protein_missingness of 0.5 indicates that a protein must have data for at least 50% of samples in the specified group to be used in the imputation algorithm and to be imputed.

Details

Methods options and descriptions:

- zero: replace missing values with 0
- median : replace missing values with global median per protein
- median_half : replace missing values with 1/2 global median per protein
- mean: replace missing values with global mean per protein
- min_samp: replace missing values with sample minimum per protein
- group_mean: replace missing values with mean per group, e.g. group (example: ROI) for each protein
- knn: imputation based on k-nearest neighbors, with proteins as neighbors, based on data from all samples across all groups. NOTE: There will still be NA values if the protein is not expressed in this group.
- group_knn: imputation based on k-nearest neighbors, with proteins as neighbors, based on data from specified group (e.g. ROI, tissue). NOTE: There will still be NA values if the protein is not expressed in this group.
- spatial_knn: imputation based on k-nearest neighors in space

Imputation will have significant impact on the outcome of statistical anlayses and further downstream investigations as well as vary depending on the chosen imputation method. We can not recommend a single universally best imputation method applicable for different types of omics data.

Value

An SPE with an 'imputed' assay with the appropriate imputation called

Examples

```
data(pancMeta)
data(protMeta)
data(smallPancData)
# We can put all samples into the same object (for statistical power)
pooledData <- dplyr::bind_cols(smallPancData)
pooled.panc.spe <- convert_to_spe(pooledData,
  pancMeta,
  protMeta,
  feature_meta_colname = "pancProts",
  sample_id = ""
)
# we can try two imputation methods and compare the difference
res <- impute_spe(pooled.panc.spe, method = "mean")
res2 <- impute_spe(pooled.panc.spe, method = "group_mean",
                   group_colname = "Image")
mean(assay(res, "imputed") - assay(res2, "imputed"), na.rm = TRUE)
```

oneKSoilsCoords — *Soil sample coordinates for the sample information*

Description

Soil sample coordinates for the sample information

Usage

```
data(oneKSoilsCoords)
```

Format

An object of class `data.frame` with 127 rows and 3 columns.

oneKSoilsMeta — *Soil samples from the 1000 soils project Row names indiciate KEGG Orthologs, Column names indicate locations Soil samples metadata for the sample information*

Description

Soil samples from the 1000 soils project Row names indiciate KEGG Orthologs, Column names indicate locations Soil samples metadata for the sample information

Usage

```
data(oneKSoilsMeta)
```

Format

An object of class `data.frame` with 142 rows and 21 columns.

pancMeta — *Here is the data to be included in the spammR package*

Description

Sample metadata for the `pancData` file.

Usage

```
data(pancMeta)
```

Format

An object of class `data.frame` with 63 rows and 15 columns.

References

pancpaper

protMeta *Protein metadata for general uniprot data, including the pancProts column for the panc data*

Description

Protein metadata for general uniprot data, including the pancProts column for the panc data

Usage

```
data(protMeta)
```

Format

An object of class `data.frame` with 6662 rows and 6 columns.

Source

Uniprot

retrieve_metaspace_data *Retrieve data objects from metaspace*

Description

`retrieve_metaspace_data`Collects data from http://metaspace2020.org by project id and returns it in SpatialExperiment object

Usage

```
retrieve_metaspace_data(
  project_id = "2024-02-15_20h37m13s",
  fdr = 0.2,
  assay_name = "lipids",
  sample_id = "sample",
  rotate = FALSE,
  drop_zeroes = TRUE,
  y_offset = 0,
  x_offset = 0
)
```

Arguments

`project_id`	Identifier for project
`fdr`	FDR value above which to collect data
`assay_name`	Name of assay to include in object
`sample_id`	Name of sample

rotate	Set to TRUE if x and y coordinates need to be swapped
drop_zeroes	Set to TRUE to drop zeroes and set the values to NAs. If False the zeroes will be included in the image data.
y_offset	Number of pixels to adjust y based on image
x_offset	Number of pixels to adjust x based on image

Details

This relies on underlying python code and environment to run, including the metaspace2020 package, the numpy package, and the pandas package.

Value

a SpatialExperiment object with the metaspace data and coordinates.

Examples

```
rdat <- retrieve_metaspace_data(project_id = "2024-02-15_20h37m13s",
                                fdr = 0.2,
                                assay_name = 'lipids',
                               sample_id = 'sample')
head(rowMeans(assay(rdat,'lipids')))
```

---

smallPancData — *Pancreatic proteomics measurements from Gosline et al. proteomics study. Row names indicate individual proteins, columns indicate sample identifiers described in* pancMeta. *Individual values show un-transformed abundance values for each protein. List of proteomic measurements in a list sorted from individual regions of the pancreas. A subset of the data at https://figshare.com/articles/dataset/spammR_Files/29250962*

Description

Pancreatic proteomics measurements from Gosline et al. proteomics study. Row names indicate individual proteins, columns indicate sample identifiers described in pancMeta. Individual values show un-transformed abundance values for each protein. List of proteomic measurements in a list sorted from individual regions of the pancreas. A subset of the data at https://figshare.com/articles/dataset/spammR_Files/

Usage

```
data(smallPancData)
```

Format

An object of class list of length 3.

Source

pancpaper

spatial_heatmap *Plot data in heatmap together with image*

Description

spatial_heatmap() Creates a spatial heatmap for a given feature in a SpatialExperiment object (spe) and provides the option to label each sample in the x-y plane

Usage

```
spatial_heatmap(
  spe,
  feature,
  feature_type = NA,
  assay_name = "proteomics",
  plotBackground_img = TRUE,
  sample_id,
  image_id,
  spatial_coord_type = "",
  spot_size_name = c("spot_width", "spot_height"),
  metric_display = "Protein expression",
  plot_log = FALSE,
  label_column = NA,
  fill_colors = c("slateblue", "darkgreen", "goldenrod"),
  sample_label_color = "white",
  sample_label_size = 1.75,
  plot_title = NULL,
  title_size = 10,
  show_na = FALSE,
  interactive = FALSE
)
```

Arguments

spe	SpatialExperiment (SPE) object
feature	Name of the feature in the spe object whose values are to plotted in the spatial heat map. This should be a row name in rowData(spe)
feature_type	Example: "GeneName". Default is whatever identifier is used in rownames. The name of feature_type must be present as a column in rowData(spe)
assay_name	Name of assay in the spe object that contains data to be plotted
plotBackground_img	Boolean (TRUE or FALSE) to indicate whether a background image should be plotted. Default is FALSE. If TRUE, the parameters image_boundaries and image_sample_id must be specified, and the image data must be present in the spe object, under imgData(spe)
sample_id	The names of the background image's sample_id fields in the spe object; this provides a unique identifier for the background image to be used for plotting (if there are multiple images under imgData(spe)) and only plots samples associated with the specified image sample_id. Example: c("Image0","Raw_noMarkings"). Image data stored under the spe object can be viewed by imgData(spe)

`image_id` The name of the background image image_id. Together with the `sample_id` this provides a unique imge to plot.

`spatial_coord_type` Position type for the given spatial coordinates of samples in spe. Current options are: "topleft_corner", "topright_corner". Default is blank, which assumes coordinates are bottom right.

`spot_size_name` Is a vector of length 2 describing the columns to find width and height of each spot.

`metric_display` Legend title for spatial heatmap. If this parameter is not specified, legend title defaults to "Protein expression"

`plot_log` Boolean to indicate if data should be log-transformed before plotting. Defaults to FALSE

`label_column` Column in colData(spe) to be used for labeling grid squares. If not specified, default is no labels.

`fill_colors` List of colors to use to provide gradient for heatmap plotting

`sample_label_color` Color to be used for labels of samples/grid squares. Default is white.

`sample_label_size` Font size for sample labels. Default is 1.75.

`plot_title` Title to be given to the spatial heatmap. Default is "Spatial signature for XYZ" where XYZ is the name of the specified feature

`title_size` Font size of title

`show_na` Set to TRUE if you want to plot NA values (will be grey) otherwise they will not be plotted

`interactive` Boolean value (TRUE/FALSE) indicating whether the plot should have interactive mouse hovering. If not specified, this defaults to TRUE. Note: grid squares can only be labeled when interactive = FALSE due to current ggplotly limitations.

Details

Assumes that every sample (column) in assay(spe) has a corresponding x,y coordinate given in spatialCoords(spe)

Assumes that colData(spe) and spatialCoords(spe) are provided in the same order of samples (rows).

Assumes that the spe object contains background image data, stored in imgData(spe) if plotBackground_img is specified to be TRUE below.

Default values are specified for function parameters 'metric_display', 'label_column' and 'interactive,' if the user doesn't specify those.

Future versions of spatialPlot_feature() to include additional options for spatial_coord_type: "center", bottomleft_corner", "bottomright_corner"

Value

A spatial heatmap of the chosen feature

Examples

```
data(pancMeta)
data(smallPancData)
data(protMeta)
img0.spe <- convert_to_spe(smallPancData$Image_0,
  pancMeta,
  protMeta,
  feature_meta_colname = "pancProts",
  image_files = system.file("extdata", "Image_0.png", package = "spammR"),
  spatial_coords_colnames = c("x_pixels", "y_pixels"),
  sample_id = "Image0",
  image_ids = "with_grid"
)
res = spatial_heatmap(img0.spe,
        feature='INS',
        sample_id='Image0',
        image_id='with_grid',
        feature_type='PrimaryGeneName',
        label_column='IsletOrNot', interactive=FALSE)
```

---

spatial_network *Plot network of correlated features*

Description

`spatial_network()` Creates a network for a given SpatialExperiment object (spe) based on the correlation of features. Will return a fully connect graph

Usage

```
spatial_network(
  spe,
  assay_names,
  feature_names = NULL,
  query_feature = NULL,
  target_features = list(),
  method = "spearman"
)
```

Arguments

`spe`	SpatialExperiment (SPE) object
`assay_names`	Name of assay in the spe object that contains data to be plotted. If only one it will focus on features in the first assay. If more than one it will compute correlations across all features.
`feature_names`	will describe which value to use for node names. IF missing will use rownames
`query_feature`	A single feature to query for correlation. If there are no `target_features` will build a network between this feature all other nodes. If missing, correlation will be computed between all nodes in `target_features`.

target_features
: List of features to compute correlation across. If query_feature is not NULL, will compute correlation only between these features and the query_feature. Otherwise, it will compute correlation between all features in this list. DEFAULT: all features in spe.

method
: to use for correlation, default is spearman

Details

Using the feature values, the algorithm computes the correlation between them across space, filtes by lowest_thresh (in the interest of compute time) and then selects the most correlated edges using the top_edges parameter.

Only the largest connected component is plotted, but the entire graph is returned as an tidygraph object.

Value

a tidygraph object that can be used for plotting or analysis

Examples

```
data(pancMeta)
data(smallPancData)
data(protMeta)
img0.spe <- convert_to_spe(smallPancData$Image_0,
  pancMeta,
  protMeta,
  feature_meta_colname = "pancProts",
  image_files = system.file("extdata", "Image_0.png", package = "spammR"),
  spatial_coords_colnames = c("x_pixels", "y_pixels"),
  sample_id = "Image0",
  image_ids = "with_grid"
)
#get 10 most variable features
vp <- sort(apply(assay(img0.spe),1,var),decreasing = TRUE)[1:10]
##now call the network
res <- spatial_network(img0.spe,'proteomics', target_features=names(vp))
```

---

spat_reduce *Reduce a high resolution image measurement (origin) to a lower resolution measurement (target)*

Description

spat_reduce() Creates a single SpatialExperiment object with all measurements mapped to the same coordinate space for plotting and analysis

Usage

```
spat_reduce(
  spe_target,
  spe_origin,
  origin_assay = "lipids",
  target_dims = c("spot_width", "spot_height"),
  replace_nas = FALSE,
  stat = "mean"
)
```

Arguments

spe_target	Low resolution image measurements
spe_origin	High-resolution image measurements to be mapped to target
origin_assay	Name of assay in origin assay to be reduced
target_dims	Names of width and height of target spots, respectively
replace_nas	Set to TRUE if we want to replace NA values with zeroes. This will increase the number of data points in common for correlation calculations.
stat	Statistic used to summarize regions, choices are currently 'mean' and 'median'

Details

This function allows us to map two distinct omics measurements to the same coordinate system. It takes on 'origin' measurement and summarizes the values (typically via mean) in the ROI in a 'target' measurement so that the source and target values can be compared.

Value

A copy of the spe_target object with the `altExp` slot containing a reduced version of the origin

---

split_spe *Divide SPE object into smaller objects by column features*

Description

`split_spe()` splits an SPE object into smaller objects by a column feature. Ideally this should be in the SpatialExperiment class or some utility package.

Usage

```
split_spe(spe, split_colname, assay_name = NULL)
```

Arguments

spe	SpatialExperiment object containing spatial omics data and spatial diffex results
split_colname	Column of rowData that maps to gene set
assay_name	Optional name of assay to use in split data

Value

A list of SpatialExperiment objects containing a subset of the data

Examples

```
data(smallPancData)
data(pancMeta)
data(protMeta)
pooledPanc <- dplyr::bind_cols(smallPancData)
panc.spe <- convert_to_spe(pooledPanc, pancMeta, protMeta,
          feature_meta_colname = "pancProts")
split_list <- split_spe(panc.spe, split_colname = "Image")
```

---

volcano_plot *Create a volcano plot from differential expression data*

Description

`volcano_plot()` generates a volcano plot to visualize results from differential expression analysis stored in a data frame. Use this function directly if differential expression results are in a separate data frame and not an spe object Helper function for volcanoPlot_DiffExSpe.R; use volcanoPlot_DiffExSpe.R if differential expression results are in an spe object. -log10(p-value) vs. log2(fold change)

Usage

```
volcano_plot(
  diffex.spe,
  logFC_colname,
  pval_colname,
  pval_corrected = TRUE,
  log_transformed = TRUE,
  title,
  thresh = 0.05,
  sigLabel_colname
)
```

Arguments

`diffex.spe`	A `SpatialExperiment` object containing differentail expression results in rowData
`logFC_colname`	column name in differenital expression results that represents log10 fold change
`pval_colname`	column name in differential epxression results that represents the p-value to be plotted
`pval_corrected`	Boolean indicating whether pval_colname represents the corrected p-value or not
`log_transformed`	set to TRUE if data is log transformed
`title`	title for the plot

`thresh` Threshold for p-value to be used to annotate significant results on the plot

`sigLabel_colname`
Either a a vector with labels or a string (Example: "Gene") that is the column name in differential expression results that should be used for labeling significant results on the plot.

Value

ggplot Volcano plot

Examples

```
data(smallPancData)
data(pancMeta)
data(protMeta)
pooledData <- dplyr::bind_cols(smallPancData)
pooled.panc.spe <- convert_to_spe(pooledData,
  pancMeta,
  protMeta,
  feature_meta_colname = "pancProts",
  sample_id = ""
)
diffex.spe <- calc_spatial_diff_ex(pooled.panc.spe,
  category_col = "IsletOrNot"
)
volcano_plot(diffex.spe,
  sigLabel_colname = "PrimaryGeneName",
  title = "Islet vs nonIslet"
)
```

Index

∗ **datasets**
oneKSoilsCoords, 14
oneKSoilsMeta, 14
pancMeta, 14
protMeta, 15
smallPancData, 16
∗ **internal**
spammR-package, 2

calc_centers, 3
calc_diff_pairwiseSamples, 4
calc_missingness, 4
calc_spatial_diff_ex, 5
convert_to_spe, 6

distance_based_analysis, 8

enrich_gradient, 9
enrich_ora, 11

impute_spe, 12

oneKSoilsCoords, 14
oneKSoilsMeta, 14

pancMeta, 14
protMeta, 15

retrieve_metaspace_data, 15

smallPancData, 16
spammR *(*spammR-package*)*, 2
spammR-package, 2
spat_reduce, 20
spatial_heatmap, 17
spatial_network, 19
split_spe, 21

volcano_plot, 22