Package: multipointR (via r-universe)

June 18, 2026

Title A package to compare intensities of point patterns across samples with spatial parametric models

Version 0.99.0

Description `multipointR` is a package to compare the distribution of cells in an image or cross images with point process models. On a single image level point process models (`ppm`) model the spatial distribution of a cell type point pattern as a function of spatial covariates while accounting for natural spacing of cells. The main model class considered in `multipointR` are inhomgoeneous Gibb's point process models. Across multiple images, users can either compare multiple univariate `ppm` models in a for loop or fit one joint model across all images with `mppm`. `multipointR` provides an interface between `SpatialExperiment` and `SpatialFeatureExperiment` objects and let's users flexibly define their own `ppm`/`mppm` models with R's formula interface.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports SummarizedExperiment, methods, SpatialExperiment, spatstat.geom, spatstat.model, spatstat.explore, formula.tools, mgcv, dplyr, ggplot2, reformulas, S4Vectors

biocViews Software, Spatial, Transcriptomics

Suggests knitr, BiocStyle, patchwork, SpatialFeatureExperiment, rmarkdown, SpatialDatasets, sosta, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 4.1.0)

VignetteBuilder knitr

Config/pak/sysreqs libmagick++-dev gsfonts libicu-dev libssl-dev zlib1g-dev

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

Date/Publication 2026-06-18 11:24:33 UTC

RemoteUrl https://github.com/BiocStaging/multipointR

RemoteRef HEAD

RemoteSha f9a3e7ab9d3686d8a370fac351811aad737b0386

Contents

`.dfToppp` *Convert SpatialExperiment object to ppp object*

Description

Convert SpatialExperiment object to ppp object

Usage

```
.dfToppp(df, marks = NULL, continuous = FALSE, window = NULL)
```

Arguments

`df`	`DataFrame`; x and y coordinates from the corresponding SpatialExperiment and the ColData
`marks`	`character`; the column with the labels e.g. cell types
`continuous`	`logical`; indicating whether the marks are continuous defaults to FALSE
`window`	`owin`; An observation window of the point pattern of class.

Value

ppp; object for use with `spatstat` functions

Examples

```
# retrieve example data from Keren et al. (2018)
spe <- SpatialDatasets::spe_Keren_2018()
speSub <- subset(spe, , imageID == "6")
dfSub <- .speToDf(speSub)
pp <- .dfToppp(dfSub, marks = "cellType")
```

`.fitSharedModelAcrossImages` *fit a shared univariate Models for all n images simultaneously*

Description

fit a shared univariate Models for all n images simultaneously

Usage

```
.fitSharedModelAcrossImages(
  spe,
  model = "ppm",
  imageId,
  imageLs = NULL,
  marks,
  formula,
  family = spatstat.model::dppGauss(),
  threshold = NULL,
  interaction = "Fiksel",
  cellspacing = NA,
  lambda = NULL,
  ncores = 1,
  verbose = TRUE,
  ...
)
```

Arguments

`spe`	`SpatialExperiment`; object subset to a single image
`model`	`character`; the `spatstat.model` function to use for computation. Typically one of `ppm`, `kppm` or `dppm`.
`imageId`	`character`; column in the colData of the `SpatialExperiment` object specifying the image ID
`imageLs`	`list`; a list specifying the subset of all images. If `NULL` then all images are considered
`marks`	`character`; the column with the labels e.g. cell types
`formula`	`formula`; the formula to pass to the `ppm` function

`family`	`detpointprocfamily`; Family to use in the point process model. One of `dppGauss`, `dppMatern`, `dppCauchy`, `dppBessel` or `dppPowerExp`
`threshold`	`numeric`; a threshold to apply on the minimum number of points a point pattern needs to have to fit a `ppm` model to it.
`interaction`	`character`; Formula specifying whether to fit a `Hardcore`, `Strauss`, `StraussHard` or `Fiksel` process to the data
`cellspacing`	`numeric` how much spacing should be accounted for in the Hardcore process due to the cell body. If this is not provided, the cell spacing parameter is estimated from the data
`lambda`	`im` or `NULL`; offset of the intensity to include in the model to account for the underlying inhomogeneity. If this is not user provided, will be estimated as inhomogeneous intensity via diggle correction
`ncores`	`numeric`; the number of cores to used for parallel processing
`verbose`	`logical`; whether to print informations on the fitting
`...`	other parameters passed on to `mppm` model from `spatstat.model`

Value

list`; result from a ppmmodel in`spatstat.model‘

Examples

```
spe <- SpatialDatasets::spe_Keren_2018()

out <- .fitSharedModelAcrossImages(spe = spe,
                imageId = "imageID",
                imageLs = list("1", "2"),
                marks = "cellType",
                formula = as.formula("Keratin_Tumour ~ distfun(CD8_T_cell)"),
                threshold = 10)
```

`.fitSingelModelsPerImage`

fit n univariate Models for all n images

Description

fit n univariate Models for all n images

Usage

```
.fitSingelModelsPerImage(
  spe,
  model = "ppm",
  imageId,
  imageLs = NULL,
  marks,
  formula,
  family = spatstat.model::dppGauss(),
  threshold = NULL,
  interaction = "StraussHard",
  cellspacing = NA,
  lambda = NULL,
  ncores = 1,
  verbose = TRUE,
  ...
)
```

Arguments

`spe`	`SpatialExperiment`; object subset to a single image
`model`	`character`; the `spatstat.model` function to use for computation. Typically one of `ppm`, `kppm` or `dppm`.
`imageId`	`character`; column in the colData of the `SpatialExperiment` object specifying the image ID
`imageLs`	`list`; a list specifying the subset of all images. If `NULL` then all images are considered
`marks`	`character`; the column with the labels e.g. cell types
`formula`	`formula`; the formula to pass to the `ppm` function
`family`	`detpointprocfamily`; Family to use in the point process model. One of `dppGauss`, `dppMatern`, `dppCauchy`, `dppBessel` or `dppPowerExp`
`threshold`	`numeric`; a threshold to apply on the minimum number of points a point pattern needs to have to fit a `ppm` model to it.
`interaction`	`character`; Formula specifying whether to fit a `Hardcore`, `Strauss`, `StraussHard` or `Fiksel` process to the data
`cellspacing`	`numeric` how much spacing should be accounted for in the Hardcore process due to the cell body. If this is not provided, the cell spacing parameter is estimated from the data
`lambda`	`im` or `NULL`; offset of the intensity to include in the model to account for the underlying inhomogeneity. If this is not user provided, will be estimated as inhomogeneous intensity via diggle correction
`ncores`	`numeric`; the number of cores to used for parallel processing
`verbose`	`logical`; whether to print informations on the fitting
`...`	other parameters passed on to `dppm` model from `spatstat.model`

Value

`mppm` object of the shared fit across all images

Examples

```
spe <- SpatialDatasets::spe_Keren_2018()

out <- .fitSingelModelsPerImage(spe = spe,
                imageId = "imageID",
                imageLs = list("1", "2"),
                marks = "cellType",
                formula = as.formula("Keratin_Tumour ~ distfun(CD8_T_cell)"),
                threshold = 10)
```

`.speToDf` *Transform a SpatialExperiment into a dataframe*

Description

Transform a SpatialExperiment into a dataframe

Usage

```
.speToDf(spe)
```

Arguments

`spe`	`SpatialExperiment`; object subset to a single image

Value

`DataFrame`; x and y coordinates from the corresponding SpatialExperiment and the colData

Examples

```
# retrieve example data from Keren et al. (2018)
spe <- SpatialDatasets::spe_Keren_2018()
speSub <- subset(spe, , imageID == "6")
dfSub <- .speToDf(speSub)
```

`defineInteractionModel` *define the Interaction Model*

Description

define the Interaction Model

Usage

```
defineInteractionModel(interaction, cellspacing, response, data)
```

Arguments

`interaction`	`character`; Formula specifying whether to fit a `Hardcore`, `Strauss`, `Fiksel` or `StraussHard` process to the data
`cellspacing`	`numeric` how much spacing should be accounted for in the interaction process due to the cell body. If this is not provided, the cell spacing parameter is estimated from the data as the minimum nearest neighbour distance divided by $n(n+1)$ as done in `spatstat.model::Hardcore`. In the case of `StraussHard`, only the Strauss interaction radius can be user-provided, the Hardcore interaction is always estimated from the data.
`response`	`character`; the response for the process, the lhs of the formula object
`data`	`list` the interaction model for the Gibbs process

Value

object of class `interact`

Examples

```
spe <- SpatialDatasets::spe_Keren_2018()
speSub <- subset(spe, , imageID == "5")
formula = as.formula("Keratin_Tumour ~ distfun(CD8_T_cell)")
#define the response
response <- as.character(formula.tools::lhs(formula))

#deparse the Formula and extract the data
out <- deparseFormula(spe = speSub,
                      response = response,
                      formula = formula,
                      marks = "cellType",
                      lambda = NULL,
                      threshold = 10)

interactionModel <- defineInteractionModel(interaction = "Hardcore",
                                           cellspacing = NA,
                                           response = response,
```

```
                              data = out[["data"]])
```

`deparseFormula` *deparse the Formula for spatstat and extract the relevant data*

Description

deparse the Formula for spatstat and extract the relevant data

Usage

```
deparseFormula(spe, formula, response, marks, lambda, threshold)
```

Arguments

`spe`	`SpatialExperiment`; object subset to a single image
`formula`	`formula`; the formula to pass to the ppm function
`response`	`character`; the response for the process, the lhs of the formula object
`marks`	`character`; the column with the labels e.g. cell types
`lambda`	`im` or `NULL`; offset of the intensity to include in the model to account for the underlying inhomogeneity. If this is not user provided, will be estimated as inhomogeneous intensity via diggle correction
`threshold`	`numeric`; a threshold to apply on the minimum number of points a point pattern needs to have to fit a ppm model to it.

Value

named `list` with both the updated formula and the data for fitting

Examples

```
spe <- SpatialDatasets::spe_Keren_2018()
speSub <- subset(spe, , imageID == "5")
formula = stats::as.formula("Keratin_Tumour ~ distfun(CD8_T_cell)")
#define the response
response <- as.character(formula.tools::lhs(formula))

#deparse the Formula and extract the data
out <- deparseFormula(spe = speSub,
                      response = response,
                      formula = formula,
                      marks = "cellType",
                      lambda = NULL,
                      threshold = 10)
```

`fitModel` *Fit point process models to one FOV of a* `SpatialExperiment` *object*

Description

Fit point process models to one FOV of a `SpatialExperiment` object

Usage

```
fitModel(
  spe,
  model = "ppm",
  marks,
  formula,
  family = spatstat.model::dppGauss(),
  threshold = NULL,
  interaction = "Fiksel",
  lambda = NULL,
  cellspacing = NA,
  ...
)
```

Arguments

`spe`	`SpatialExperiment`; object subset to a single image
`model`	`character`; the `spatstat.model` function to use for computation. Typically one of `ppm`, `kppm` or `dppm`.
`marks`	`character`; the column with the labels e.g. cell types
`formula`	`formula`; the formula to pass to the `ppm` function
`family`	`detpointprocfamily`; Family to use in the point process model. One of `dppGauss`, `dppMatern`, `dppCauchy`, `dppBessel` or `dppPowerExp`
`threshold`	`numeric`; a threshold to apply on the minimum number of points a point pattern needs to have to fit a `ppm` model to it.
`interaction`	`character`; Formula specifying whether to fit a `Hardcore`, `Strauss`, `Fiksel` or `StraussHard` process to the data
`lambda`	`im` or `NULL`; offset of the intensity to include in the model to account for the underlying inhomogeneity. If this is not user provided, will be estimated as inhomogeneous intensity via diggle correction
`cellspacing`	`numeric` how much spacing should be accounted for in the interaction process due to the cell body. If this is not provided, the cell spacing parameter is estimated from the data as the minimum nearest neighbour distance divided by $n(n+1)$ as done in `spatstat.model::Hardcore`. In the case of `StraussHard`, only the Strauss interaction radius can be user-provided, the Hardcore interaction is always estimated from the data.
`...`	other parameters passed on to `ppm` model from `spatstat.model`

Value

`list`; result from a dppm model in `spatstat.model`

Examples

```
spe <- SpatialDatasets::spe_Keren_2018()
speSub <- subset(spe, , imageID == "5")

mdl <- fitModel(spe = speSub,
                marks = "cellType",
                formula = as.formula("Keratin_Tumour ~ distfun(CD8_T_cell)")
)
```

`fitModelAcrossImages` *Fit point process models across all images in the* `SpatialExperiment` *object*

Description

Fit point process models across all images in the `SpatialExperiment` object

Usage

```
fitModelAcrossImages(
  spe,
  model = "ppm",
  imageId,
  imageLs = NULL,
  marks,
  formula,
  family = spatstat.model::dppGauss(),
  threshold = NULL,
  interaction = "Fiksel",
  cellspacing = NA,
  lambda = NULL,
  sharedModel = TRUE,
  ncores = 1,
  verbose = TRUE,
  ...
)
```

Arguments

`spe`	`SpatialExperiment`; object subset to a single image
`model`	`character`; the `spatstat.model` function to use for computation. Typically one of ppm, kppm or dppm.

`imageId`	`character`; column in the colData of the `SpatialExperiment` object specifying the image ID
`imageLs`	`list`; a list specifying the subset of all images. If `NULL` then all images are considered
`marks`	`character`; the column with the labels e.g. cell types
`formula`	`formula`; the formula to pass to the `ppm` function
`family`	`detpointprocfamily`; Family to use in the point process model. One of `dppGauss`, `dppMatern`, `dppCauchy`, `dppBessel` or `dppPowerExp`
`threshold`	`numeric`; a threshold to apply on the minimum number of points a point pattern needs to have to fit a `ppm` model to it.
`interaction`	`character`; Formula specifying whether to fit a `Hardcore`, `Strauss`, `StraussHard` or `Fiksel` process to the data
`cellspacing`	`numeric` how much spacing should be accounted for in the Hardcore process due to the cell body. If this is not provided, the cell spacing parameter is estimated from the data
`lambda`	`im` or `NULL`; offset of the intensity to include in the model to account for the underlying inhomogeneity. If this is not user provided, will be estimated as inhomogeneous intensity via diggle correction
`sharedModel`	`logical`; whether or not to fit one univariate model per image (n models in total) or estimate one shared univariate model across all images.
`ncores`	`numeric`; the number of cores to used for parallel processing
`verbose`	`logical`; whether to print informations on the fitting
`...`	other parameters passed on to `dppm` model from `spatstat.model`

Value

`list`; result from a `ppm` model in `spatstat.model` or a `mppm` model

Examples

```
spe <- SpatialDatasets::spe_Keren_2018()

mdl <- fitModelAcrossImages(spe = spe,
              imageId = "imageID",
              imageLs = list("1", "2"),
              marks = "cellType",
              formula = as.formula("Keratin_Tumour ~ distfun(CD8_T_cell)"),
              threshold = 10)
```

`mdlToDf` *Convert the list of spatial models to a dataframe*

Description

Convert the list of spatial models to a dataframe

Usage

```
mdlToDf(mdlLs, imageCovariates = c("imageID"))
```

Arguments

`mdlLs`	`list`; A list of all the models fit with e.g. `ppm`
`imageCovariates`	`list`; A list of all the covariates on the image level to add to the model dataframe

Value

DataFrame of the model coefficients

Examples

```
spe <- SpatialDatasets::spe_Keren_2018()

mdlLs <- fitModelAcrossImages(spe = spe,
                imageId = "imageID",
                imageLs = list("1", "2"),
                marks = "cellType",
                sharedModel = FALSE,
                formula = as.formula("Keratin_Tumour ~ distfun(CD8_T_cell)"),
                threshold = 10)

mdlDf <- mdlToDf(mdlLs = mdlLs,
                 imageCovariates = c("imageID",
                                     "tumour_type"))
```

`plot.multipointRppm` *Plot mulitpointR* `ppm` *objects*

Description

Plot mulitpointR `ppm` objects

Usage

```
## S3 method for class 'multipointRppm'
plot(x, type = "trend", ...)
```

Arguments

`x`	`ppm`; a model fit with `spatstat.model::ppm` or the wrapper `multipointR::fitModel`
`type`	`character`; the type to plot, one of "intensity" or "trend"
`...`	further arguments passed to geom_raster

Value

a `ggplot2` object of the model trend/intensity surfance

Examples

```
spe <- SpatialDatasets::spe_Keren_2018()
speSub <- subset(spe, , imageID == "5")

mdl <- fitModel(spe = speSub,
                marks = "cellType",
                formula = as.formula("Keratin_Tumour ~ distfun(CD8_T_cell)")
)
plot(mdl)
```

`speToPPP` *convert a SpatialExperiment object into a point pattern*

Description

convert a SpatialExperiment object into a point pattern

Usage

```
speToPPP(spe, marks)
```

Arguments

`spe`	`SpatialExperiment`; object subset to a single image
`marks`	`character`; the column with the labels e.g. cell types

Value

A ppp object for use with any `spatstat` package

Examples

```
spe <- SpatialDatasets::spe_Keren_2018()
speSub <- subset(spe, , imageID == "6")
pp <- speToPPP(speSub, mark = "cellType")
```

Index